Mathematicians, Spies and Hackers

Coding and encryption

Everything is mathematical

Mathematicians, Spies and Hackers

Mathematicians, Spies and Hackers

Coding and encryption

Joan Gómez

Everything is mathematical

To my son Vicenç

Published by RBA Coleccionables, S.A.
c/o Hothouse Developments Ltd
91 Brick Lane, London, E1 6QL

Localisation and cover design: Windmill Books Ltd.
Photographic credits: age fotostock, Aisa, Album, Corbis, Getty Images, iStockphoto

ISSN 2050-649X

Printed in Spain

Table of Contents

Preface

A common game in any school playground is to invent a special alphabet for sending and receiving secret messages. The effort devoted to these childhood codes has much more to do with the enthusiasm of the would-be spies than the threat of any third party snooping on the information being transmitted. In the adult world, however, such unwanted interest clearly exists, and the confidentiality of many communications is extraordinarily important.

Once limited to the activities of a political and social elite, the arrival of the information age has made codes and ciphers essential to the smooth functioning of society as a whole. This book attempts to explain the history of secret codes from the point of view of the most qualified of guides: mathematics.

Cryptography, that is, the art of writing in code, appeared alongside writing itself. Although the Egyptians and Mesopotamians made use of encryption methods, the first to apply themselves to it fully were the early Greeks and the Romans, aggressive cultures for whom communicating in secret was a key element of their military success. Such secrecy brought about new kinds of adversaries – those who declare themselves the keepers of the secret, the cryptographers, and those who hope to reveal it, the cryptanalysts or code breakers. This is always a battle carried out behind the scenes, which, over time may give the advantage temporarily to one side or the other, but never reaches a decisive victory. In the 8th century, for example, the Arab sage, Al-Kindi, came up with one deciphering tool known as frequency analysis, which looked as though it could foil anyone writing in code. The eventual response (it took centuries to appear) of encoders was the polyalphabetic cipher. It, too, seemed to be a decisive weapon – until a more sophisticated code-breaking system, devised by an English genius in the privacy of his study, once again turned the advantage. Ever since, the principal weapon employed by one side or the other has been mathematics, from statistics to modular arithmetic, by way of number theory.

This encoding and deciphering battle reached a turning point with the appearance of the first encryption machines, followed not long after by machines devoted to decoding. The first programmable digital computer, named Colossus, was invented and built by the British to crack coded messages from Enigma, the German encoding device.

With the explosion of computing power, codes acquired a leading role in the transmission of information beyond the traditional considerations of secrecy. The

universal language of modern society does not use letters or ideograms, but two digits – 0 and 1. This is the binary code.

Which side benefited the most from the arrival of the new technologies, the cryptographers or the cryptanalysts? Is security still possible in this age of viruses, data theft and supercomputers? The answer to the second question is very much yes, and again we have to thank mathematics, in this case prime numbers and their particular characteristics. How long will this momentary dominance of the secret last? The answer to this question will take us to the furthest frontiers of contemporary science, to the theories of quantum mechanics, where astounding paradoxes will mark the end of this exciting journey through the mathematics of security and secrecy.

This book ends with a bibliography for those who wish to go deeper into the world of encoding and cryptography, and the index will aid in the search.

Chapter 1
How Secure is Information?

Cryptography: the art of writing or solving codes
Oxford Dictionary

The desire to create a message that can only be understood by the sender and its recipient – and is meaningless to any other person – is arguably as ancient as writing itself. In fact, there exists a series of "nonstandard" hieroglyphics that are more than 4,500 years old, although we do not know with any certainty whether they represented an attempt to conceal information or were instead playing a part in some kind of ritual. We know more about a Babylonian tablet dated around 2,500 BC. It contains words with the first consonant removed and employs some unusual variants of characters. Investigations have revealed that the text describes a method of producing glazed ceramic, which leads us to conclude that it was engraved by a merchant or perhaps a potter who was concerned to protect trade secrets from competitors.

With the spread of writing and trade came the birth of great empires, which in turn were engaged in frequent border disputes. Cryptography and the secure transmission of information became a priority for governments as well as merchants. Today, in the information age, the need to protect the integrity of communication and to maintain an appropriate level of privacy is more important than ever. There is scarcely any flow of information that is not encoded in one way or another. The purpose of the code is to make it easier to send. For example, text is converted into the binary language (a numerical system using just 0 and 1) intelligible to a computer. Once encoded, most of this information can be protected from anyone that might intercept it. In other words, the code needs to be encrypted. Finally, the legitimate recipient has to be able to decipher the message. Encoding, encrypting and deciphering are the basic steps in the "dance of information" that is repeated millions of times per second of every minute, of every hour of every day. And the music that accompanies this dance is none other than mathematics.

Codes, ciphers and keys

Cryptographers use the term encode in a slightly different sense from the rest of us. For them, encoding is a method of writing in code that consists of substituting one word for another. On the other hand, using encryption or a cipher involved substituting letters or other single characters. With the passage of time, the latter form has become so prevalent that it has become synonymous with "writing in code". However, if we follow the more scholarly interpretation, the correct term for the second method would be to encrypt (or decrypt, in the case of the reverse process) a message.

Let's imagine we are sending a secure message "ATTACK". We could do so in two basic ways: substituting the word (code), or substituting some or all of the letters that make up the word (cipher). A simple way to encode a word is to translate it into a language that the potential eavesdroppers won't know, whereas with encryption it would be sufficient, for example, to substitute each letter with another located elsewhere in the alphabet. In each case, it is necessary that the receiver knows the procedure that has been employed to encode or encrypt the message, or our message will be useless. If we had already agreed with the recipient that we would use one method or the other – translate it to another language or substitute each letter with another – all we would need to do would be to inform him or her of the targeted language or the number of places we have moved forward in the alphabet to substitute each letter. In an encrypted example, if the recipient gets the message "CVVCEM" and knows that we have moved each letter forward two places, he can easily reverse the process and decrypt the message.

THE BINARY CODE

For a computer to understand and process information, it must be translated from the language in which it is written into the so-called binary language. This language consists of two digits only: 0 and 1. The binary expression for 0–10 in the decimal system is shown in the table on the right.

Consequently, the decimal number 9,780 would be expressed, in binary code, as 10011000110100

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10

TO TRANSLATE OR TO DECRYPT?

Translations of text written in a language using an unknown character set can be approached as a general problem of decryption. The translation can be seen as the unknown text already translated into our language, and the encrypting algorithm would be the grammatical rules and syntax of the original language. The techniques used for both tasks – to translate and decrypt – have many similarities In both cases the same condition needs to be met the sender and the receiver must, at the least, share a common language That is why the translation of texts written in lost languages, such as the Egyptian hieroglyphic or Linear B, was impossible until a way of corresponding them to a known language was found In both cases, this was Ancient Greek The picture above is of a tablet found in Crete written in Linear B.

The distinction we have established between the encryption rule (the system being applied) and the parameter of encryption (a variable instruction that is specific to each message or a group of messages) is extremely useful because a potential spy would need to know both to decipher the message Thus the spy could know that the key to the cipher is to substitute each letter with the corresponding letter a specific number, x, places further forward in the alphabet However, if he does not know what x is, he will have to try all possible combinations one for each letter of the alphabet. In this example, the cipher is very simple and to exhaust all the possibilities – what is known as brute-force decryption – is not particularly laborious However, in the case of more complex techniques, this type of code breaking, or cryptanalysis, is practically impossible, by hand at any rate Moreover, the interception and deciphering of messages are both generally subject to important time restrictions. The information has to be obtained and understood before it becomes useless or widely known by others.

HOW MANY KEYS ARE REQUIRED?

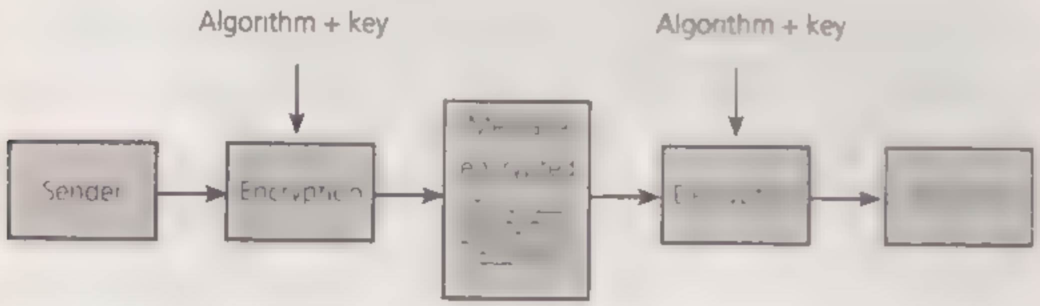
What is the minimum number of keys needed in a system with two users? Three? Four? For two users to communicate with each other secretly, only one code or key is necessary. In the case of three users, three are needed: one for the communication between A and B, another for the pair A and C, and a third for B and C. Similarly, four users would require six keys. Thus, to generalize, for n users we would need as many keys as there are combinations of pairs of n users, that is

$$\binom{n}{2} = \frac{n(n-1)}{2}.$$

So a relatively small system of 10,000 interconnected users would require 49,995,000 keys. In the case of a world population of six billion individuals, the number is dizzying: 17,999,999,997,000,000,000.

The general rule of encryption is often termed the encryption algorithm, while the specific parameter used to cipher or encode the message is termed the key (In the ciphering example on page 10, for example, the key is 2. Each original letter is replaced by another two places further on in the alphabet). Obviously a great number of keys are possible for every encryption algorithm, and so knowing the algorithm alone can be a good as useless unless we also know the key used to encrypt it. Since the keys are generally easier to change and to disseminate, it seems logical to concentrate on keeping the keys most secret in order to maintain the security of an encryption system. This principle was established at the end of the 19th century by the Dutch linguist Auguste Kerckhoffs von Nieuwenhof, and is thus known as Kerckhoffs' principle.

To summarise what we have presented to this point, we can set out a general system of encryption defined by the following elements:



That is, a sender and a recipient of the message, an encryption algorithm and a defined key that allows the sender to cipher the message and the receiver to decipher it. Later, we will see how this diagram has been modified in recent times because of the changing nature and function of keys, but for the time being we will stick to this diagram.

Private keys and Public keys

Kerckhoffs' principle establishes the key as the fundamental element in the security of any cryptographic system. Until relatively recently, the keys of a sender and a receiver in all conceivable cryptographic systems needed to be identical or at least symmetrical, that is, they needed to be used for both the encryption and decryption of a message. The key was, therefore, a secret shared by the sender and the recipient, and thus the cryptographic system in use was always vulnerable, so to speak, from both sides. This type of cryptography, which is dependent on a key shared by the sender and the receiver, is known as a private key.

All cryptographic systems invented by humans since the beginning of time, irrespective of the algorithm used and its complexity, shared this characteristic.

HOW MANY KEYS ARE REQUIRED?... PART 2

As we have seen on page 12, classical cryptography required an enormous number of keys. However, in a public cryptographic system any two users who exchange messages only need four of them: their respective public and private keys. In this case n users require $2n$ keys.

Making the key the same for the recipient and the sender seems to be common sense. After all, how can one person encode a message according to one code, and a second decipher it according to another and hope that the meaning of the text is retained? For thousands of years this possibility was considered a logical absurdity. However, as we shall see in more detail later, just five decades ago the absurd became entirely possible, and is now a ubiquitous part of codes.

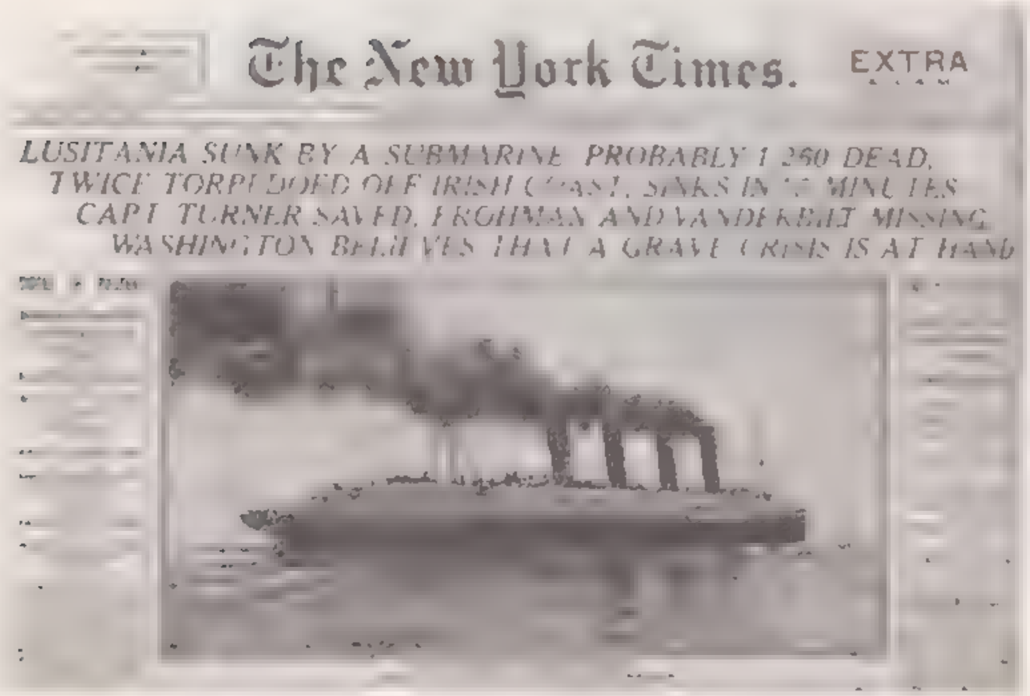
Nowadays, encryption algorithms used in the majority of communications consist of at least two keys: a secret, private one, as was already customary, and a public one known by everyone. The transmission mechanism is as follows: the sender gets the public key of the recipient to whom he wishes to send the message and uses it to encrypt the message. The receiver takes his private key and uses it to decipher the received message. Moreover, this system possesses an extremely important additional advantage: neither the sender nor the recipient need to have got together in advance to agree on any of the keys involved, so the security of the system is very much tighter than was possible before. This completely revolutionary form of encryption is known as public key, and forms the basis of the security underlying today's communication networks.

Mathematics is at the root of this revolutionary technology. In effect, as we shall explain in detail later on, modern cryptography sits on two foundations. The first is modular arithmetic, while the second is number theory — particularly the part concerning the study of prime numbers.

The Zimmermann telegram

Cryptography is one of the areas of applied mathematics in which the contrast between the pristine clarity of the underlying theory and the murky consequences of its implementation are most apparent. After all, the destiny of entire nations depends on the success or the failure of maintaining secure communications. One of the most spectacular examples of how cryptography changed the course of history occurred almost a century ago, in what became known as the Zimmermann telegram affair.

On May 7, 1915, with half of Europe engaged in bloody conflict, a German U-boat torpedoed the transatlantic passenger liner *Lusitania*, which was sailing under the British flag near the coast of Ireland. The result was one of history's most infamous massacres: 1,198 civilians, 124 of whom were American, lost their lives. The news enraged public opinion in the United States, and the government of President



The New York Times. EXTRA

LUSITANIA SUNK BY A SUBMARINE, PROBABLY 1,260 DEAD;
TWICE TORPEDOED OFF IRISH COAST; SINKS IN 15 MINUTES;
CAPT. TURNER SAVED, FROHMAN AND VANDERBILT MISSING;
WASHINGTON BELIEVES THAT A GRAVE CRISIS IS AT HAND

How The New York Times *reported the sinking of the* Lusitania

Woodrow Wilson warned his German counterparts that any similar act would lead to the immediate entry of the United States into the war on the Allied side. In addition, Wilson demanded that German submarines surface before carrying out any attack so as to avoid the sinking of further civilian ships. The tactical advantage of the U-boat force was therefore seriously compromised.

In November, 1916, Germany appointed Arthur Zimmermann, a man with a reputation for diplomacy, as its new foreign minister. The news was welcomed by the United States press, who saw his appointment as a good omen for relations between Germany and the USA.

In January, 1917, less than two years after the tragedy of *Lusitania*, and with the conflict at its peak, the German ambassador to Washington, Johann von Bernstorff, received the following coded telegram from Zimmermann, with instructions to deliver it in secret to his counterpart in Mexico, Heinrich von Eckardt:

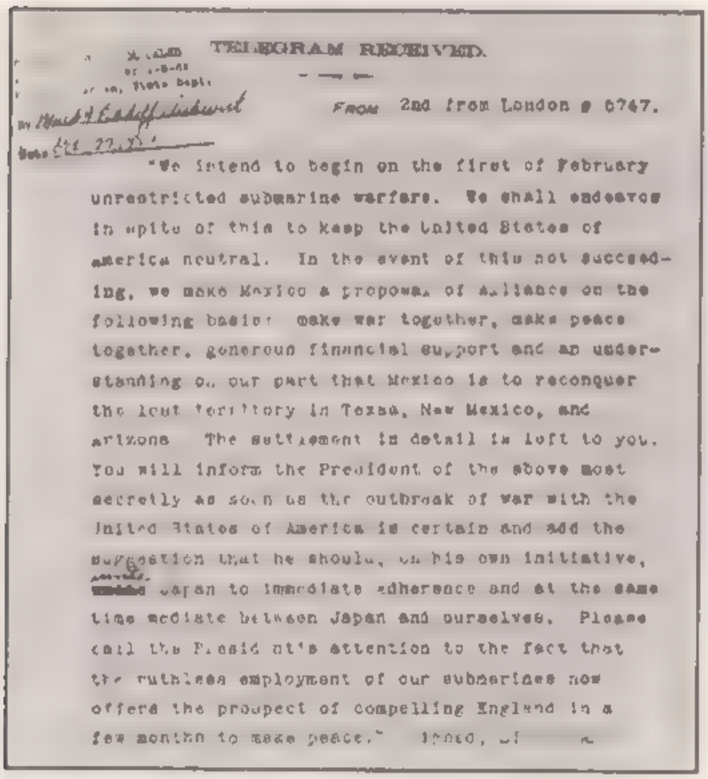
> "We intend to begin on the first of February unrestricted submarine warfare. We shall endeavour in spite of this to keep the United States of America neutral.
>
> In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous

> financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you [von Eckardt].
>
> You will inform the President [of Mexico] of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, invite Japan to immediate adherence and at the same time mediate between Japan and ourselves.
>
> Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace."

If it had been made public, the certain consequence of this telegram would have been the outbreak of war between Germany and the United States. Although Kaiser Wilhelm II knew this would be inevitable once submarines operated without surfacing before an attack, he hoped that by then the United Kingdom would have capitulated and therefore there would be no conflict for the United States to join. Barring this circumstance, the active threat of Mexico along the southern border of the United States could equally dissuade that country from entering another conflict many miles away. Mexico, however, was going to need a certain amount of time to organise its forces. Therefore it was vital that Germany's intentions remain secret long enough for the submarine warfare to tip the balance of the conflict in Germany's favour.

Room 40 gets to work

The British government, however, had other plans. Shortly after the start of the war they had cut the undersea telegraphic cables that connected Germany directly with the Western Hemisphere, so any electronic communications had to go via cables that the British could intercept. The United States, in an attempt to bring about a negotiated end to the conflict, had been allowing Germany to continue transmitting diplomatic messages. As a result, Zimmermann's message was received intact by the German delegation in Washington DC.

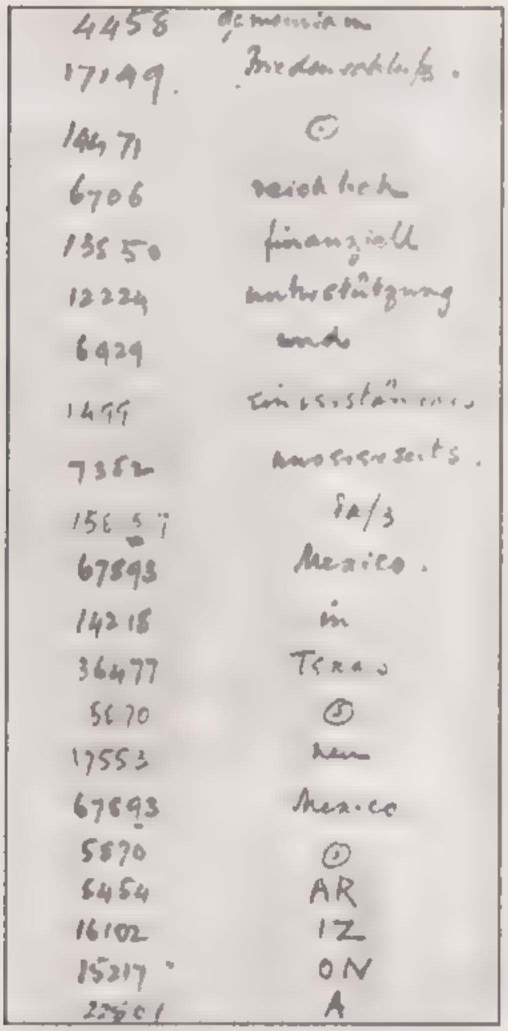


TELEGRAM RECEIVED.

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, ~~invite~~ Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, Z

Zimmermann's telegram (top) forwarded by the German ambassador in Washington DC, Heinrich von Eckardt, to his counterpart in Mexico with the deciphered version of the same telegram below it


4458 gemeinsam
17149 Friedensschluss.
14471
6706 reichlich
13850 finanziell
12224 unterstützung
6929 und
14991 einverständnis
7382 unsererseits.
15857
67893 Mexico.
14218 in
36477 Texas
5670
17553 neu
67893 Mexico
5870
5454 AR
16102 IZ
15217 ON
22801 A

Part of the British decoding of Zimmermann's telegram. In the lower part can be seen how the Germans, lacking a code for the word "Arizona", encoded it in sections AR, IZ, ON, A.

The British government sent the intercepted message to its code-breaking department, known as Room 40.

The Germans had used their normal foreign ministry encryption algorithm and had used a cipher known as 0075, which the experts of Room 40 had already partially broken. The algorithm in question involved the substitution of words (encoding), as well as letters (ciphering), a practice similar to that used in another of the encrypting tools used at that time by the Germans, the cipher ADFGVX, which we will examine in more detail later.

The British did not take long to decipher the telegram, although they were reluctant to show it to the Americans right away. There were two reasons for this. First, the secret telegram had been transmitted under the diplomatic cover provided by

the United States to German messages, a privilege that the British had ignored. Second, if the telegram was made public, the German government would immediately know that its codes had been compromised and would change its system of encryption. Therefore, the British decided to tell the Americans that the intercepted and decrypted version was the one forwarded by Eckardt to Mexico, and so convince the Germans that the telegram had been intercepted, already decrypted, in Mexico.

At the end of February, Wilson's government leaked the contents of the telegram to the press. Some members of the press – particularly the newspapers belonging to the Hearst group, which was anti-war and pro-German – were sceptical at first. However, by mid-March, Zimmermann publicly admitted to being the author of the controversial message. A little over two weeks later, on April 6, 1917, the US Congress declared war on Germany, a decision that would have far-reaching consequences for Europe and the world.

Although extraordinary in its time, Zimmermann's telegram is just one of the historical landmarks in which cryptography has played an essential role. Throughout this book we will see many other examples, scattered throughout the centuries and from all cultures. Even so, we can be almost certain that we do not know about many of the most crucial events. By its very nature, the history of cryptography is a secret history.

Chapter 2

Cryptography from Antiquity to the 19th Century

As we have already noted, cryptography is an ancient discipline, probably as ancient as written communication itself. However, it is not the only possible method for transmitting information in secret. After all, every text has to have a medium, and if we make the medium invisible to everyone except the recipient, we will have accomplished our objective. The technique of concealing the existence of the message itself is called steganography, and it probably originated around the same time and for the same reasons as cryptography.

Steganography

The Greek scholar Herodotus, considered one of the world's greatest historians, mentions in his famous chronicle of the war between the Greeks and the Persians in the 5th century BC, two curious instances of steganography that reveal a considerable amount of ingenuity. In the first example, contained in Book III of Herodotus's *History*, Histiaeus, the tyrant of Miletus, commanded a man to shave his head. He then wrote the message that he wanted to send on the man's scalp and waited for his hair to grow back. The man was then sent to his destination, Aristagoras' camp. Safely there, the messenger explained the ploy to Aristagoras and shaved his hair off again, revealing the long awaited message. The second example, if true, is of greater historical importance because it allowed Demaratus, a Spartan king exiled in Persia, to warn his compatriots of an imminent invasion by the Persian king, Xerxes. Herodotus takes up the story in Book VII:

> "The fact was that Demaratus could not warn them just like that, so he had the following idea: he took a pair of [writing] tablets, scraped off the wax and wrote the king's plans on the wooden surface of the tablets. He then covered them with melted wax, thus concealing the message.

> In this way the tablets, being apparently blank, would cause no trouble with the guards stationed along the road.
>
> When the tablets finally reached Lacedaemon (Sparta), the Lacedaemonians couldn't figure out the secret until, as I understand it, Gorgo [...] suggested that they scrape the wax off the tablets because they – she indicated – would find a message engraved on the wood beneath."

A steganographic device that has stood the test of time is invisible ink. Celebrated in thousands of stories and films, the materials used – lemon juice, plant sap, and even human urine – are generally of organic origin and have a high carbon content. Therefore, they tend to darken when exposed to moderately high temperatures, such as the heat from a candle flame.

Steganography's usefulness is beyond dispute, although it is utterly unfeasible when dealing with large numbers of communications. Moreover, used on its own the technique has a significant flaw: if the message were to be intercepted, the contents would be immediately apparent. For this reason steganography is principally employed as a complement to cryptography, a means of strengthening the security of top secret transmissions.

We can deduce from the examples given that armed conflict has been a great driver for the secure transmission of information. This being so, it is not surprising that a martial people such as the Spartans – if we believe Herodotus, already masters at steganography – would also be pioneers in the development of cryptography.

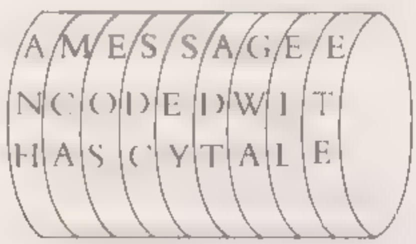
Transposition cryptography

In the conflict between the Spartans and the Athenians for control of the Peloponnese, frequent use was made of long strips of paper wrapped around a cylinder, known as a *scytale*. A message was then written on the coiled paper. Even if the technique used (that is, the encryption algorithm) was known by the enemy, if the exact dimensions of the scytale were not known, anyone intercepting the message would find it extremely difficult to decipher its meaning. The thickness and length of the scytale were, in essence, the key to the encryption system. When the paper strip was unwound, the message became illegible.

WITH TINY LETTERS

During the years of the Cold War [illegible] spy thrillers [illegible] the protagonists sending detailed messages by way of [illegible] the naked eye [illegible] microfilm. The technique was born [illegible] agents used a steganographic technique [illegible] microdot [illegible] photograph of a brief text reduced to the size of a [illegible] [illegible] typographical symbols within an innocuous text

In the illustration below, the message (M) to be transmitted is "A message encoded with a scytale", but the unwound strip of paper displays the incomprehensible gibberish (C): "anh mca eos sdc sey adt gwa eil ete."



M = A MESSAGE ENCODED WITH A SCYTALE

C = ANH MCA EOS SDC SEY ADT GWA EIL ETE

Using a scytale employs a cryptographic technique known as transposition, where the letters in the message are reordered. To get an idea of the power of this method, consider the simple example of transposing just three letters A, O and R. A quick test with no calculations necessary reveals that they can be reordered in up to six different ways: AOR, ARO, OAR, ORA, ROA and RAO.

In abstract terms, the process is as follows: once one of the three possible letters is placed first, allowing for three different arrangements, we are left with two letters that can in turn be reordered in two different ways for a new total of 3x2 = 6 arrangements. In the case of a somewhat longer message of, for example, 10 letters, the number of possible arrangements is now

A MANUAL FOR YOUNG LADIES

The *Kama Sutra* is a lengthy manual that deals, among other things, with the knowledge that a woman needs in order to be a good wife. Written around the 4th century BC by the Brahmin Vatsyayana, it recommends up to 64 different skills, including music, cooking and chess. Number 45 is of particular interest to us because it deals with the art of secret writing, or *mlecchita vikalpa*. The learned author recommends several methods, including the following: divide the alphabet in half and pair the resulting letters at random. In this system, each pairing of letters represents a key. For example, one of them could be the following.

A	S	C	D	N	F	G	X	J	I	K	Z	M
E	O	P	Q	R	B	T	U	W	V	H	Y	L

To write the secret message one would thus have to substitute every A in the original text with E, P with C, J with W, etc., and vice versa.

$10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$. Such an operation is expressed by the mathematical notation $10!$ and produces a total of 3,628,800. In general terms, for n number of letters, there are $n!$ different ways to reorder them. So, a message of a modest 40 letters would produce so many ways to reorder the letters that it would be practically impossible to decipher by hand. Have we perhaps found the perfect cryptographic method?

Not entirely. In effect, a random algorithm of transposition offers a higher level of security, but what is the key that allows it to be deciphered? The randomness of the process is both its strength and its weakness. Another encryption method was needed that would generate keys that were simple, easy to remember and to transmit, without sacrificing large amounts of security. So began the search for the perfect algorithm, and the first successes were achieved by the Roman emperors.

To Caesar what is Caesar's

Veni, vidi, vici (I came, I saw, I conquered).
Julius Caesar

Substitution ciphers developed in parallel with transposition ciphers. Unlike transposition, strict substitution exchanges one letter for another, or any type of symbol.

Unlike transposition, substitution does not draw on just the letters that appear in the message. In transposition, the letter changes its position, but maintains its role; the same letter has the same meaning in the original message and in the ciphered message. In substitution, the letter maintains its position but changes its role (the same letter or symbol has one meaning in the original message and another in the ciphered message). One of the first known substitution algorithms is the so-called Polybius cipher, in honour of the Greek historian Polybius (2[illegible]–2[illegible]), who left us a description of it. His method is developed in full in the Appendix.

Approximately 50 years after the Polybius cipher, in the first century BC, another substitution cipher appeared, known by the generic name of Caesar's cipher because Julius Caesar was one of its most infamous practitioners. Caesar's cipher is one of the best studied in the field of cryptography and it is extremely useful because it illustrates the principles of modular arithmetic, one of the mathematical foundations of writing in code.

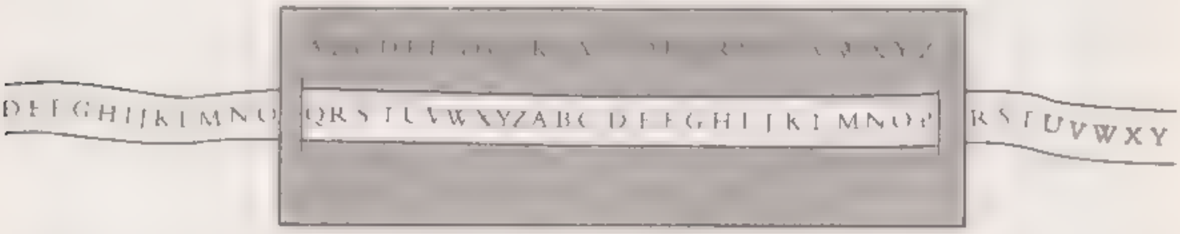
Caesar's cipher operates by replacing each letter of the alphabet with another one from a fixed number of positions down the alphabet. According to the great historian Suetonius in his *The Twelve Caesars*, Julius Caesar coded his personal correspondence with a substitution algorithm of this type: each letter of the original message was substituted by another that followed three positions further

GAIUS JULIUS CAESAR (100–44 BC)

Caesar (right) was a soldier and statesman whose dictatorship would end the Roman Republic. After serving as magistrate in Hispania Ulterior, he joined two other powerful people of the period, Pompey and Crassus, and with them formed the First Triumvirate, validated by the marriage of Julia, Caesar's daughter, to Pompey. The three divided up the Roman empire. Crassus got command of the eastern provinces, Pompey remained in Rome, and Caesar assumed the military command of Cisalpine Gaul and the Proconsulship of Narbonese Gaul. At this time, the war against the Gauls began. It lasted eight years and culminated in the Romans conquering Gaulish territory. From there, Caesar marched back to the imperial capital with his victorious legions and installed himself as sole dictator.

down the alphabet: the letter A was substituted by D, B by E, and so on W became Z, and so X,Y and Z reverted to A, B and C.

The encoding and decoding of a message encrypted in this way could be carried out with a simple device like the one below:

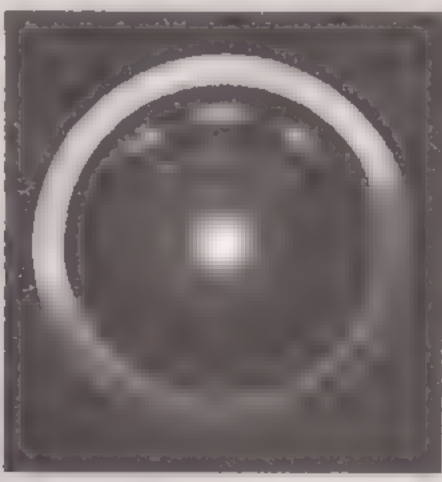


Now we will examine the process in greater detail. In the table below, we see the starting alphabet and the transformation caused by Caesar's cipher of substituting the letter three positions further down the alphabet of 26 letters (the upper row shows the original alphabet and the lower row shows the ciphered alphabet).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

FILM CODES

In the classic science fiction film *2001: A Space Odyssey* (1968), directed by Stanley Kubrick and based on a story by Arthur C. Clarke, a spacecraft's supercomputer, called HAL 9000, is endowed with consciousness and becomes insane, attempting to kill the human crew. Now take Caesar's Cipher with a key of B and treat the word "HAL" as a message encrypted with that code. We see that the letter H corresponds to the letter I; the A to the letter B, and the L to the letter M, in other words, "IBM", at the time the largest computer manufacturer in the world. Was the film making a comment about the dangers of artificial intelligence or the pitfalls of unregulated commercial power? Or was it just a coincidence?



The all-seeing eye of HAL 9000 from the film 2001: A Space Odyssey.

When the two alphabets, the original (or plaintext) and the ciphered are arranged in this way, to encrypt any message it is simply a question of substituting the letters of one with those of the other. The key to the cipher is named after the letter that corresponds to the encrypted value for A (the first letter of the original alphabet). In this case, it is the letter D. The classic expression "AVE CAESAR" ("Hail Caesar") would be encrypted as "DYH FDHVDU". Conversely, if the encrypted message is "WUHH", then the decrypted or plaintext message is "TREE". In the case of the Caesar code just described, a cryptanalyst who had intercepted the message and knew the algorithm being used but not the key, would have to try all possible orderings until he found a message that made sense. To do this he would need to explore, at the most, the total number of keys or displacements. With an alphabet of n letters, n possible displacements produce n number of codes.

16 = 4. Modular arithmetic and the mathematics of Caesar's cipher

$16 = 4^2$ and $2 = 14^2$. This is not a mistake, nor is it some strange numbering system. The operation of a Caesar cipher can be formulated with a tool that is very common in mathematics and even more so in cryptography – modular arithmetic, sometimes called clock arithmetic. This technique had its origins in the work of the Greek mathematician Euclid (325–265 BC), and it is one of the fundamentals of modern information security. In this section, we will introduce the basic mathematical concepts related to this particular type of arithmetic.

THE FATHER OF ANALYTIC CRYPTOGRAPHY

[illegible] and the geometry of [illegible] Although mostly associated with [illegible] of the Greek mathematician [illegible] modules [illegible] constitutes [illegible] and [illegible] appeared [illegible] incidental that both [illegible] were great masters of cryptography.

Take a classic analogue clock as an example and compare it with a digital one. The analogue distribution of the hours divides the circle into 12 parts that we will write as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. The equivalent numbering of pm hours between an analogue clock and a digital one can be seen in the following table.

0	1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22	23

When, for example, we say that it is "14:00" we are also saying that it is two o'clock in the afternoon. The same principle applies in the case of the measurement of angles. A 370° angle is equivalent to a 10° angle because you have to deduct a complete 360° turn from the first value. Note that $370 = (1 \cdot 360) + 10$ and also that 10 is the remainder when 370 is divided by 360. What angle is equivalent to 750°? Deducting the relevant complete turns we find that a 750° angle is equivalent to a 30° angle. We conclude that $750 = 2 \cdot 360 + 30$ and that 30 is the remainder of dividing 750 by 360. The mathematical notations for this are:

$$750 \equiv 30 \text{ (mod. 360)}.$$

And we say that "750 is congruent with 30 modulus 360." In the case of the clock, we would write $14 \equiv 2$ (mod. 12).

We could also imagine a clock with negative numbers. In this case, what time would it be when the hand of the clock points to −7? Or, in other words, what would −7 be congruent with in modulus 12? Let us calculate this remembering that the value "0" in our 12-part clock is equivalent to "12:"

$$-7 = -7 + 0 \equiv -7 + 12 = 5.$$

CALCULATIONS WITH MODULI

How to calculate 231 in modulus 17 with a calculator?

First we divide 231 by 17 and we get 13.58823529.

[illegible] multiply the product, 13x17 = 221. In this way we do away with the decimals involved all together.

[illegible] the subtraction 231−221 = 10, thus obtaining the remainder of the division.

231 in modulus 17 is 10. This datum is expressed as 231 ≡ 10 (mod 17).

The mathematics of the calculations with our analogue 12-part clock is called arithmetic in modulus 12. In general terms, we can say that $a \equiv b$ (mod m) if the remainder of the division between a and m is b, given that a, b and m are whole numbers. The number b is equivalent to the remainder of dividing a by m. The following statements are equivalent

$$a \equiv b \text{ (mod. } m)$$
$$b \equiv a \text{ (mod. } m)$$
$$a - b \equiv 0 \text{ (mod. } m)$$
$$a - b \text{ is a multiple of } m$$

The question "What analogue time is 19 hours?" is equivalent in mathematical terms to the following question "What is 19 congruent with in modulus 12?" To answer this question we have to solve the equation

$$19 \equiv x \text{ (mod. 12).}$$

Dividing 19 by 12 we get the quotient 1 and the remainder 7, so

$$19 \equiv 7 \text{ (mod. 12).}$$

And in the case of 127 hours? We divide 127 by 12 and we get the quotient 10 and the remainder 7, therefore

$$127 \equiv 7 \text{ (mod. 12).}$$

To reiterate what we have learned so far, let's examine the following operations in modulus 7 set out below:

(1) $3+3 \equiv 6$

(2) $3+14 \equiv 3$

(3) $3 \times 3 = 9 \equiv 2$

(4) $5 \times 4 = 20 \equiv 6$

(5) $7 \equiv 0$

(6) $35 \equiv 0$

(7) $-44 = -44+0 \equiv -44+7 \times 7 \equiv 5$

(8) $-33 = -33+0 \equiv -33+5 \times 7 \equiv 2$

(1) 6 is less than the modulus, and so is unchanged

(2) $3+14=17$;	$17:7=14$ and a remainder of 3
(3) $3\times 3=9$;	$9:7=1$ and a remainder of 2
(4) $5\times 4=20$;	$20:7=2$ and a remainder of 6
(5) $7=7$;	$7:7=1$ and a remainder of 0
(6) $35=35$;	$35:7=5$ and a remainder of 0
(7) $-44=-44+0$;	$-44+(7\times 7)=5$
(8) $-33=-33+0$;	$-33+(5\times 7)=2$

MULTIPLICATION TABLE IN MODULUS 5 USING EXCEL

A multiplication table in modulus 5 would look like this:

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

It is easy to formulate this and other similar tables with only a modest knowledge of Excel spreadsheets. In the case of our example, the syntax of the Excel expressions on our computer (using our row and column positions) are shown below. The concept "remainder of dividing a number by 5" is translated into Excel language by "=remainder(number,5)". The actual instruction for finding the product of 4 times 3 in modulus 5 would be, then, "=remainder(4*3,5)", an operation that would give us the value 2. Such tables are very helpful in carrying out modular arithmetic calculations.

	1	2	3	1
=REMAINDER(B$5*$A6,5)	=REMAINDER(C$5*$A6,5)	=REMAINDER(D$5*$A6,5)	=REMAINDER(E$5*$A6,5)	=REMAINDER(F$5*$A6,5)
=REMAINDER(B$5*$A7,5)	=REMAINDER(C$5*$A7,5)	=REMAINDER(D$5*$A7,5)	=REMAINDER(E$5*$A7,5)	=REMAINDER(F$5*$A7,5)
=REMAINDER(B$5*$A8,5)	=REMAINDER(C$5*$A8,5)	=REMAINDER(D$5*$A8,5)	=REMAINDER(E$5*$A8,5)	=REMAINDER(F$5*$A8,5)
[illegible]$5*$A9,5)	=REMAINDER(C$5*$A9,5)	=REMAINDER(D$5*$A9,5)	=REMAINDER(E$5*$A9,5)	=REMAINDER(F$5*$A9,5)

What is the relationship between modular arithmetic and Caesar's cipher? To answer the question we will set out a conventional alphabet and an alphabet with a displacement of 3 letters, to which we add a numerical header corresponding to the 26 characters.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	[illegible]	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	[illegible]	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

We can see that the ciphered version of letter number x (in the plaintext alphabet) is the letter that occupies the position $x + 3$ (also in the plaintext alphabet). So it is important to find a transformation in which each numerical value is assigned the same value displaced by three units and take the result in modulus 26. Note that 3 is the key of the cipher. So its function is defined as

$$C(x) = x + 3 \text{ (mod. 26)},$$

where x is the unciphered value and $C(x)$ is the ciphered value. It is sufficient to substitute the letter by its numerical equivalent and apply the transformation. Let us take as an example the message "PLAY" and let us encode it

The P would be 15, $C(15) = 15 + 3 \equiv 18$ (mod. 26), which corresponds to S
The L would be 11, $C(11) = 11 + 3 = 14$ (mod. 26), thus obtaining O
The A would be 0, $C(0) = 0 + 3 = 3$ (mod. 26), thus obtaining D
The Y would be 24, $C(24) = 24 + 3 = 27 \equiv 1$ (mod. 26), thus obtaining B

The message "PLAY" ciphered in a key of 3 is "SODB"

In general, if x indicates the position of the letter we wish to encode (0 for A, 1 for B, etc.), the position of the ciphered letter (denoted by $C(x)$) will be expressed by the formula

$$C(x) = (x + k) \text{ (mod. } n)$$

where n is the length of the alphabet (26 in the English alphabet) and k is the key, which transforms the ciphered message according to its value.

The deciphering of such a message involves the reverse calculations to ciphering it. In terms of our example, deciphering is equivalent to applying the inverse formula to the one used in ciphering:

$$C^{-1}(x) = (x - k) \text{ (mod. } n).$$

In the case of the message ciphered "SODB", with a Caesar cipher with a key of 3 in the English alphabet, $k = 3$ and $n = 26$, therefore

$$C^{-1}(x) = (x - 3) \text{ (mod. 26)}.$$

The process is as follows:

For S, $x = 18$, $C^{-1}(18) = 18\text{-}3 \equiv 15$ (mod 26), which corresponds to P.
For O, $x = 14$, $C^{-1}(14) = 14\text{-}3 \equiv 11$ (mod 26), by which we obtain L.
For D, $x = 3$, $C^{-1}(3) = 3\text{-}3 \equiv 0$ (mod.26), obtaining the A.
For B, $x = 1$, $C^{-1}(1) = \text{-}2+26 \equiv 24$ (mod.26), obtaining the Y.

The message "SODB" ciphered in Caesar's cipher with a key of 3 corresponds, as we already know, to the plaintext "PLAY."

To conclude this first foray into the mathematics of cryptography, we can establish a new transformation, known as an affine cipher, which generalises Caesar's cipher. The transformation is defined as:

$$C_{(a,b)}\ (x) = (a \cdot x + b) \text{ (mod. } n)$$

with a and b being two whole numbers smaller than the number (n) of letters in the alphabet. The greatest common denominator (gcd) between a and n has to be 1 [$\gcd(a,n) = 1$], because otherwise there would be the possibility of ciphering the same letter in different ways, as we shall see later on. The key of the cipher is determined by the pair (a,b). Caesar's cipher with a key of 3 would, then, be an affine cipher with the values of $a = 1$ and $b = 3$.

The general affine ciphers like these offer greater security than a conventional Caesar cipher. Why? As we have seen, the key of an affine cipher is pairs of numbers (a,b). In the case of a message written in an alphabet of 26 letters and encrypted by means of an affine cipher, a and b can adopt any value between 0 and 25. The

GREATEST COMMON DENOMINATOR (GCD)

The greatest common de[illegible] algorithm. Th[illegible] between [illegible] the [illegible] numbers. For example,

$$\gcd(48,30)?$$

48 is divided by 30 and we get the remainder 18 and the quotient 1.

30 is divided by 18 and we get the remainder 12 and the quotient 1.

18 is divided by 12 and we get the remainder 6 and the quotient 1.

12 is divided by 6 and we get the remainder 0 and the quotient 2

We have completed the algorithm.

The $\gcd(48,30)$ is 6.

If the gcd $(a,n) = 1$, we say that a and n are coprime

[illegible] establishes that for two integers a and n larger than 0, there are integers k and q such that gcd $(a,n) = ka + nq$

number of keys possible in this system of encryption with an alphabet of 26 letters is, therefore, 25x25 = 625. We observe that the number of keys for an alphabet of n letters is n times greater than that of Caesar's cipher. The increase is considerable, but it is still susceptible to deciphering by brute force.

Playing spies

Under what conditions is it possible to decipher a message encrypted with an affine cipher, whether as the intended recipient or as a spy? We will explore this question using a simple example of a cipher for an alphabet of six letters

0	1	2	3	4	5
A	B	C	D	E	F

The text will be encrypted with the affine cipher $C(x) = 2x + 1 \pmod 6$.

The A is ciphered according to $C(0) = 2 \times 0 + 1 \equiv 1$ (mod. 6), which corresponds to B.
The B is ciphered according to $C(1) = 2 \times 1 + 1 \equiv 3$ (mod. 6), which corresponds to D.
The C is ciphered according to $C(2) = 2 \times 2 + 1 \equiv 5$ (mod. 6), which corresponds to F
The D is ciphered according to $C(3) = 2 \times 3 + 1 = 7 \equiv 1$ (mod. 6), which corresponds to B
The E is ciphered according to $C(4) = 2 \times 4 + 1 = 9 \equiv 3$ (mod. 6), which corresponds to D
The F is ciphered according to $C(5) = 2 \times 5 + 1 = 11 \equiv 5$ (mod. 6) which corresponds to F

The proposed affine cipher encrypts the messages "ABC" and "DEF" in the same way and the original message is lost. What has happened?

If we work with a cipher expressed as $C_{(a,b)}(x) = (ax + b)$ (mod n), we can decipher the message unequivocally only if the $\gcd(a,n) = 1$. In our example, $\gcd(2,6) = 2$ and therefore fails this restriction.

The mathematical operation of deciphering is equivalent to finding the unknown x given a numerical value y in modulus n.

$$C_{(a,b)}(x) \equiv (ax + b) \equiv y \text{ (mod. } n)$$
$$(ax + b) \equiv y \text{ (mod. } n)$$
$$ax \equiv y - b \text{ (mod. } n).$$

In other words, we are seeking a value a^{-1} (the inverse of a), which satisfies $a^{-1}a = 1$, such that

$$a^{-1}ax \equiv a^{-1}(y - b) \text{ (mod. } n)$$
$$x \equiv a^{-1}(y - b) \text{ (mod. } n).$$

Consequently, to decipher successfully we have to calculate the inverse of a number a in modulus n and, in order to avoid wasting time, we need to know in advance if there really is such an inverse. An affine cipher $C_{(a,b)}(x) = (ax + b)$ (mod n), will have an inverse if, and only if, the $\gcd(a,n) = 1$.

In the case of the affine cipher in the example, $C(x) = 2x + 1$ (mod 6), we want to know if the number a, in our case 2, has an inverse. That is, if there is a whole number u smaller than 6 such that $2 \cdot u \equiv 1$ (mod. 6). To do this we solve for all the values of the moduli (0,1,2,3,4,5):

$$2\cdot 0 = 0,\ 2\cdot 1 = 2,\ 2\cdot 2 = 4,\ 2\cdot 3 = 6 \equiv 0,\ 2\cdot 4 = 8 \equiv 2,\ 2\cdot 5 = 10 \equiv 4.$$

There is no such value, from which we conclude that 2 does not have an inverse. In reality, we already knew this since $\gcd(2,6) \neq 1$.

Let's now assume that we have intercepted a coded message "YSFMG". We know that it has been encrypted with the affine cipher in the form of $C(x) = 2x + 3$ and was originally written in Spanish with an alphabet of 27 letters (including an Ñ following the regular N). What is the original message? First we calculate the gcd(2,27), which is equal to 1. The original message can be deciphered! To do so we have to find the inverse function of $C(x) = 2x + 3$ in modulus 27.

$$y = 2x + 3$$
$$2x = y - 3.$$

To isolate the x we have to multiply both sides of the equation by the inverse of 2. The inverse of 2 in modulus 27 is a whole number u such that $2 \cdot u \equiv 1$ (mod. 27), that is 14, which we confirm:

$$14 \cdot 2 = 28 \equiv 1$$

Consequently,

$$x = 14(y - 3).$$

Now we can decipher the message:

The letter Y occupies position 25 and deciphered it will be $14(25 - 3) = 308 \equiv 11$ (mod. 27).

The letter that occupies position 11 in the alphabet is L.

In the case of the letter S, $14(19 - 3) = 224 \equiv 8$ (mod. 27), which corresponds to the letter I.

In the case of F, $14(5 - 3) = 28 \equiv 1$ (mod. 27), which corresponds to B.

In the case of M, $14(12 - 3) = 126 \equiv 18$ (mod. 27), which corresponds to O.

The deciphered message is the Spanish word "LIBRO" (meaning *book*).

Beyond the affine cipher

Various security systems were based for many centuries on Caesar's idea and its generalisation in the form of the affine cipher. Nowadays any cipher in which each letter of the original message is substituted by another letter that has been shifted a fixed number of places (not necessarily three) is called Caesar's cipher.

One of the greatest virtues of a good encrypting algorithm is the ability to generate a large quantity of keys. Both Caesar's cipher and the affine cipher are vulnerable to cryptanalysis because the maximum number of keys is low.

If we eliminate any restriction regarding the order of the letters of the ciphered alphabet, however, the potential number of keys increases markedly. The number of keys available to the standard 26-character (in any order) alphabet is $26! = 403{,}291{,}461{,}126{,}605{,}635{,}584{,}000{,}000$, that is 403 septillion keys. A code breaker investigating one potential key every second would take more than one billion times the expected life of the universe to exhaust all the possibilities!

A possible code with a general substitution algorithm could be the following:

(1)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
(2)	Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

Row (1) Plaintext alphabet. Row (2) Ciphered alphabet.

The first six letters of the ciphered alphabet give a clue as to the selected ordering: it corresponds to the order of the letters on a keyboard that follows the QWERTY standard. To cipher Caeser's famous comment "VENIVIDIVICI" ("I came, I saw, I conquered") with the QWERTY code, for every letter of the conventional alphabet we look for the corresponding one in the ciphered alphabet.

(1)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
(2)	Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

That would give us the following ciphered message:

CTFO CORO COEO

There is a very simple way to generate an almost inexhaustible number of codes that are easy to remember for this ciphering method. It is sufficient to agree on any *keyword* (it can even be a phrase) and place it at the beginning of the ciphered alphabet, allowing the rest of the alphabet to follow the conventional order starting with the last letter of the keyword, taking care not to repeat any letters. An example would be "JANUARY CIPHER". First we would eliminate the space and the repeated letters, thus getting the keyword "JNUYCIPHE." The resulting ciphered alphabet would be the following:

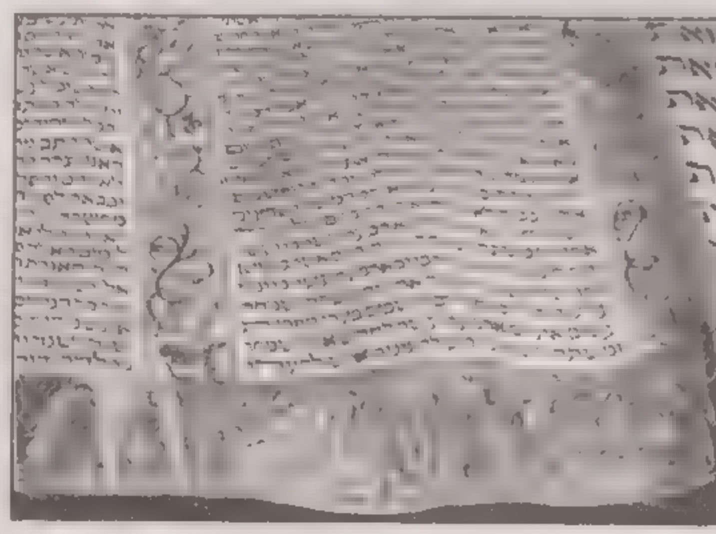
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	N	U	Y	C	I	P	H	E	F	G	K	L	M	O	Q	R	S	T	V	W	X	Z	A	B	D

The message "VENI VIDI VICI" would now be ciphered as "XCME XEYE XEUE". This system of generating codes can be arranged so that sender and receiver error are unlikely and it is simple to update. In our example, it would be enough to change the code each month – from JANUARY CIPHER to FEBRUARY CIPHER and from there to MARCH CIPHER etc. – without the communicators having to speak to each other after the code was established.

The reliability and simplicity of the keyword substitution algorithm made it the preferred encrypting system for many centuries. During that time the general consensus was that the cryptographers had the upper hand over the cryptanalysts.

CIPHERING THE WORD OF GOD

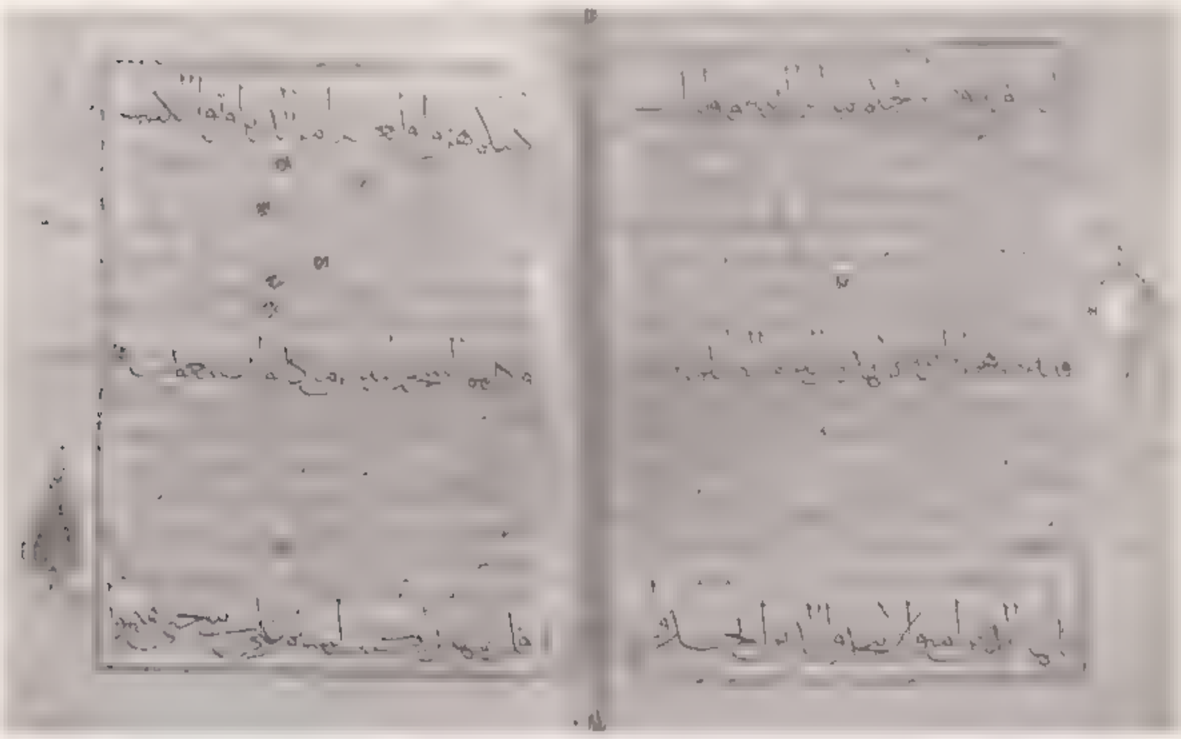
Medieval cryptanalysts believed they saw ciphers in the Old Testament, and they were not mistaken. There are several fragments of sacred texts that are encrypted with [illegible] called *Atbash*. This cipher consists in substituting any letter [illegible] distance from the end of the alphabet as from the beginning [illegible] the A is substituted by Z, B by Y, etc. In the case of the original Old Testament the substitutions are carried out with the letters of the Hebrew alphabet. So in Jeremiah (25,26) the word "Babel" is ciphered as "Sheshakh."



A Hebrew Bible from the early 18th century.

Frequency analysis

The Koran is composed of 114 chapters, each of which corresponds to one of the Prophet Muhammad's revelations. These revelations were written down during the life of the Prophet by various companions and later collected by Abu Bakr, the first caliph. Umar and Uthman, the second and third caliphs respectively, completed the project. The fragmentary nature of the original writings encouraged the birth of a branch of theology devoted to the exact dating of the different revelations. Among other dating techniques, Koranic scholars compiled the frequency of the appearance of certain words considered to be newly coined throughout the writing period. If a revelation contained enough of these newer words, it was reasonable to conclude that it was a comparatively late revelation.



14th century Koran manuscript.

This initiative turned out to be the first specific cryptanalysis tool ever invented: frequency analysis. The first person to leave a written record of this revolutionary technique was a philosopher by the name of Al Kindi, who was born in Baghdad around the year 801. Although he was an astronomer, doctor, mathematician and linguist, the occupation for which he is most remembered is that of cryptanalyst. If he was not the first, Al Kindi was certainly the most important one in history.

Very little was known about Al-Kindi's pioneering role until relatively recently. In 1987, a copy of a treatise of his entitled *On Deciphering Cryptographic Messages* surfaced in an archive in Istanbul. This contains a very succinct precis of the groundbreaking technique:

> "One way to decode a ciphered message, if we know in what language it is written, is to find a plaintext written in the same language that is sufficiently long, and then count how many times each letter appears. The letter that appears with the most frequency we will call the "first," the next most frequent we will call "second"... and so on until we have covered all the letters that appear in our text. Then we observe the coded text that we are deciphering and we classify its symbols in the same manner. We find the symbol that appears with the most frequency, and we substitute it with the "first" from our text, we do the same with the "second" and so on, until we have covered all the symbols of the cryptogram we are deciphering."

In earlier pages, he mentions that in the substitution cipher method, each letter of the original message "maintains its position but changes its role," and it is precisely this constancy of "maintaining the position" that makes it susceptible to frequency cryptanalysis. Al-Kindi's genius reversed the balance of power between cryptographers and cryptanalysts, swinging it, for a time at least, toward the eavesdroppers.

A detailed example

From greatest to least frequent, this is how letters are used in English texts: E T A O I N S H R D L C U M W F G Y P B V K J X Q Z. The percentage of appearances made by each letter is shown in the following frequency table:

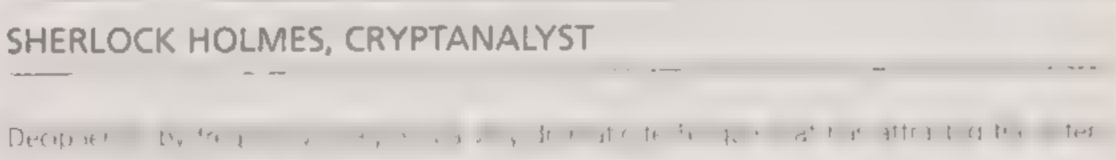
A	8.17%	H	6.09%	O	7.51%	V	0.98%
B	1.49%	I	6.97%	P	1.93%	W	2.36%
C	2.78%	J	0.15%	Q	0.10%	X	0.15%
D	4.25%	K	0.77%	R	5.99%	Y	1.97%
E	12.70%	L	4.03%	S	6.33%	Z	0.07%
F	2.29%	M	2.41%	T	9.06%		
G	2.02%	N	6.75%	U	2.76%		

If a message has been ciphered with a substitution algorithm like the ones discussed earlier, it is open to being decoded according to the relative frequency of the letters of the original message. It is enough to count the appearance of each of the ciphered letters and compare them to the frequency table of the language in which it was written. So, if the letter that appears most often in the ciphertext is, for example, J, the letter of the original message to which it most likely corresponds would be, in the case of English, an E. If the second most frequent letter is Z, the same reasoning would lead us to conclude that T is the most likely corresponding letter. The process is repeated for all the letters of the ciphertext and thus the cryptanalysis is complete.

Obviously the frequency method cannot always be applied so directly. The frequencies of the previous table are correct only on average. Short texts such as "*Visit the zoo kiosk for quiz tickets*" have a relative frequency of letters that is very different to that which characterises the language as a whole. In effect, in texts of less than

SHERLOCK HOLMES, CRYPTANALYST

Deciphering by frequency [illegible] everyday people with its ingenuity.



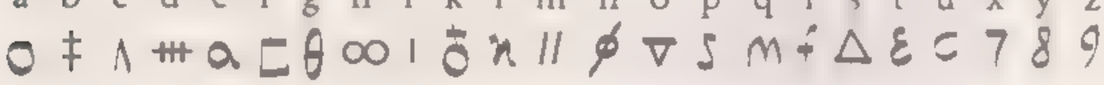
[illegible] important element of the cipher

100 characters this simple analysis is rarely useful. Frequency analysis, however, is not limited to the study of letters on their own. Although we agree that it is often unlikely that the most frequent letter in a short ciphertext is E, we can be more certain that the five most frequent letters are probably A, E, I, O and T, without knowing which corresponds to which. A and I never appear in pairs in English, while the other letters can. Moreover, it is also likely that, however short the text, the vowels tend to appear in front of and behind clusters of other letters, while the consonants tend to group with vowels or with small numbers of letters. In this way, we can perhaps differentiate the T from the A, E, I and the O. As we successfully decipher some letters, words will appear where we only need to decipher one or two characters, which will allow us to pose hypotheses on the identity of those letters. The speed with which we can decipher increases as we decipher more letters.

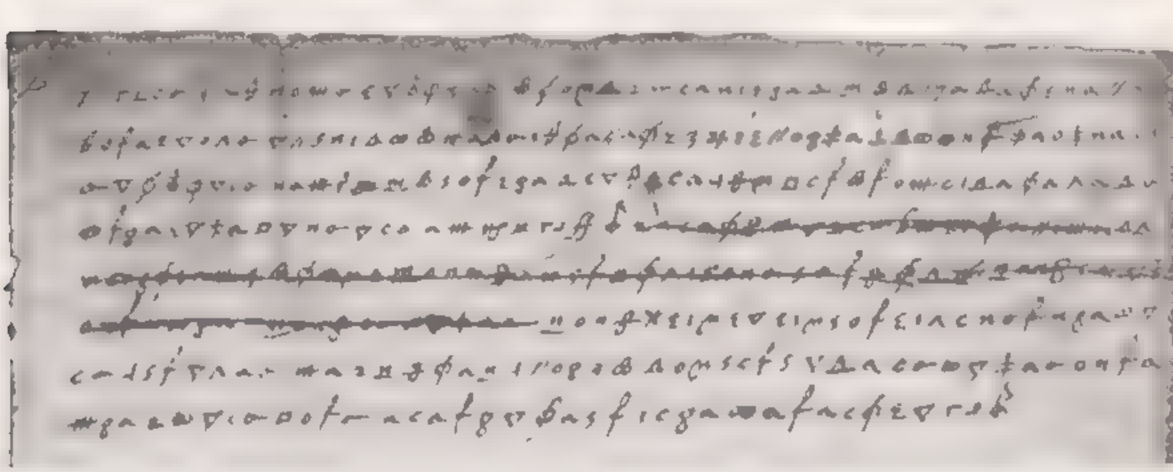
The polyalphabetic cipher

On February 8, 1587, Mary, Queen of Scots, was beheaded at Fotheringhay Castle after being found guilty of treason. The judicial proceedings leading to such a drastic sentence had demonstrated beyond doubt that Mary had been colluding with a group of Catholic aristocrats, headed by the young Anthony Babington, in a plan to assassinate Queen Elizabeth I of England and install Mary at the head of a Catholic kingdom encompassing both England and Scotland. The decisive evidence was offered by Elizabeth's counterespionage service, headed by Lord Walsingham. It was comprised of a series of letters between Mary and Babington which clearly stated that the young Scottish queen knew about the deadly plan and approved of it. The letters in question were ciphered with an algorithm that combined ciphers and codes. In other words, not only did it exchange letters with other characters, but it also employed unique symbols to refer to certain words of common usage. Mary's ciphered alphabet appears below:

a b c d e f g h i k l m n o p q r s t u x y z



Except for the fact that it used symbols instead of letters, Mary's ciphered alphabet is no different to any other used for centuries by cryptographers all over the world. The young queen and her conspirators were convinced that the cipher was secure but, unfortunately for her, Elizabeth's best cryptanalyst, Thomas Phelippes, was an expert in frequency analysis and was able to decipher Mary's letters with little difficulty. The thwarting of what came to be known as the Babington Plot sent a powerful signal to the governments and agents of all Europe: the conventional substitution algorithm was no longer secure. The cryptographers appeared impotent in the face of the power of the new deciphering tools.



A fragment of one of Mary, Queen of Scots' letters to the conspirator Anthony Babington which would eventually condemn her to death

Alberti's contribution

However, a solution to the problem posed by frequency analysis had been found more than a century before Mary's head was put on the block. The architect of the new cipher was none other than the multi-talented Renaissance scholar Leon Battista Alberti. Generally better known as an architect and mathematician who made great leaps forward in the study of perspective, in 1460 Alberti devised a system of encryption that consisted of adding a second ciphered alphabet to the first one as shown in the following table:

(1)	A	B	C	D	E	F	G	H	[illegible]	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
(2)	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
(3)	M	N	B	V	C	X	Z	L	K	J	H	G	F	D	[illegible]	A	P	O	[illegible]	[illegible]	Y	T	R	E	W	Q

Row (1) Plaintext alphabet. Row (2) Ciphered alphabet 1. Row (3) Ciphered alphabet 2

To encrypt any message whatsoever, Alberti proposed alternating the two ciphered alphabets. For example, in the case of the word "SHEEP" the cipher for the first letter would be found in the first alphabet (V), that of the second in the second (L), and so on. In our example, "SHEEP" would be ciphered as "VLHCS". The advantage of this *polyalphabetic* encryption algorithm, in comparison with the previous ones, is evident straight away – the double E from the plaintext is ciphered in two different ways, H and C. To further confuse any cryptanalyst faced with the encrypted text, the same ciphered letter represents two different letters in the plaintext. Frequency analysis, therefore, lost a large part of its usefulness. Alberti never formally set out his idea in a treatise, and the cipher was later developed independently at more or less the same time by two academics, the German Johannes Trithemius and the French Blaise de Vigenère.

De Vigenère's square

In Caesar's cipher, a monoalphabetic cipher is used, a single ciphered alphabet corresponds to the plaintext alphabet such that the same ciphered letter always corresponds to the same plaintext letter (In the classic Caesar cipher, D is always an A, E is B, and so on).

In a polyalphabetic cipher, on the other hand, a particular letter in a message can be assigned as many letters as the number of ciphered alphabets used. To encrypt a text, a different ciphered alphabet is used as one goes from one letter of the plaintext alphabet to the next. The first and most famous polyalphabetic cipher system is known as De Vigenère's square. His table of alphabets consisted of a plaintext alphabet of n letters below which appeared n ciphered alphabets, each one shifted cyclically by one letter to the left in comparison to the previous alphabet above. In other words, a square matrix of 26 rows and 26 columns arranged as shown on the next page.

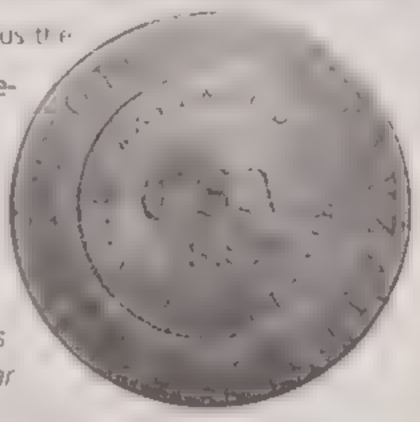
Note the symmetry in the correspondence of the letters. The pair (A,R) = (R,A), and this same relationship applies to all the letters.

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
2	B	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
3	C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
4	D	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
5	E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
6	F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
7	G	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
8	H	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
9	I	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
10	J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
11	K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
12	L	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
13	M	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
14	N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
15	O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
16	P	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
17	Q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
18	R	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
19	S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
20	T	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
21	U	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
22	V	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
23	W	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
24	X	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
25	Y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
26	Z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

We can immediately see that DeVigenère's square consists of a plaintext alphabet of n letters each one of which is transformed according to increasing parameters. So the first ciphered alphabet would serve to apply a Caesar cipher with the parameters $a = 1$ and $b = 2$, the second would be equivalent to a Caesar cipher with $b = 3$, etc. The key to DeVigenere's square consists of knowing which letters of the message are ciphered and how many rows down we go to find the corresponding ciphered letter. The simplest key consists of moving down one row for every letter of the original message.

PLAYING WITH DISKS

A practical way to implement a polyalphabetic [illegible] known as an Alberti cipher disk. These portable ciphers [illegible] with a conventional alphabet engraved [illegible] The sender can by rotating the moveable ring [illegible] alphabets as there are turns in the ring [illegible] of the alphabet being used. [illegible] an Alberti disk [illegible] analysis. To decrypt the message, the recipient only has to make [illegible] the sender. The security of this cipher, as always, depends on keeping [illegible] the arrangement of the alphabet on the moveable ring plus the number of turns effected. An Alberti disk with a single moveable ring engraved with a traditional alphabet allows for a Caesar cipher at every turn. Similar devices were used in conflicts as recent as the American Civil War, and today they can be found in children's spy games.



An Alberti disk used by the Confederates in the American Civil War

So our classic phrase "VENI VIDI VICI" would be ciphered as follows:

To cipher the first V, we find the corresponding letter in row 2: W
To cipher the E, we find the corresponding letter in row 3: G
To cipher the N, we find the corresponding letter in row 4: Q
I (row 5): M
V (row 6): A
I (row 7): O
D (row 8): K
I (row 9): Q
V (row 10): E
I (row 11): S
C (row 12): N
I (row 13): U

DIPLOMAT AND CRYPTOGRAPHER

Blaise de Vigenère was born in France in 1523. In 1549, he was sent by the French government on a diplomatic mission to Rome, where he became interested in cryptography and ciphered messages. In 1585, he wrote his seminal work, *Traicté des Chiffres* (*Treatise on Ciphers*), which describes the system of encryption to which he gave his name. This cipher system was unassailable for almost two centuries, until the Briton Charles Babbage succeeded in deciphering it in 1854. Curiously enough, this fact was not known until some time into the 20th century, when a group of scholars examined Babbage's personal notes and calculations

The original encrypted phrase would become "WGQM AOKQ LSNU." As can be immediately verified, the repeated letters in the original message disappear. However, every cryptographer's concern is to generate codes that are easy to remember, to distribute and to update. Keywords that had the same or fewer numbers of letter as the message being deciphered were used to generate shorter, easier to use De Vigenère's squares. The keyword formed the first letters in each row (see page 47), followed by the rest of the alphabet (as they appear in the full square. Then the keyword was written below the plaintext, repeating as often as is necessary. Then the letter in the keyword below each of the plaintext characters directs the cryptographer to the row in the square from which the ciphered letter is to be taken

For example, if we wish to cipher the message "BUY MILK TODAY" by means of the keyword "JACKSON":

Original message	B	U	Y	M	I	L	K	T	O	D	A	Y
[illegible]	J	A	C	K	S	O	N	J	A	C	K	S
[illegible]	K	U	A	W	A	Z	X	C	O	F	K	Q

The ciphered message is "KUAWAZXCOFKQ."

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	j	k	[illegible]	m	n	[illegible]	p	q	r	[illegible]	[illegible]	[illegible]	[illegible]	w	[illegible]	[illegible]	z	[illegible]	b	[illegible]	d	e	f	g	h	[illegible]
A	a	b	c	d	e	f	g	h	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	y	z
C	c	d	e	f	g	h	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	a	b
K	k	l	m	n	o	p	q	r	s	t	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	g	h	[illegible]	[illegible]
S	s	t	u	v	w	[illegible]	[illegible]	[illegible]	[illegible]	b	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	o	p	q	r
O	o	p	q	r	s	t	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	m	n
N	n	o	p	q	r	[illegible]	t	[illegible]	v	w	x	[illegible]	[illegible]	[illegible]	b	c	d	e	[illegible]	g	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

De Vigenère's square with the rows defined by the keyword JACKSON

As in the case of all classical encryption systems, the deciphered message of a text encrypted using De Vigenère's square is symmetrical to the ciphered message. For example, for the case of the message ciphered "WZPKGIMQHQ" with a keyword of "WINDY":

Original message	?	?	?	?	?	?	?	?	?	?
Keyword	W	I	N	D	Y	W	[illegible]	N	D	Y
Ciphered message	W	Z	P	K	G	I	M	Q	H	Q

Let's look at the first column. We are seeking to solve the unknown "?" given that (?,W) = W. To do this we look along the W row in the De Vigenère's square on page 44 until the W appears and we see which column it corresponds to; the answer is A. Next, we look for a letter "?" that verifies that (?,I) = Z and we get R, and so on. The original message is revealed as "ARCHIMEDES".

The historical importance of De Vigenère's square, which it shares in general with other polyalphabetic ciphers such as Gronsfeld's (developed at a similar time and explained in detail in the Appendix), is its resistance to frequency analysis. If the same letter could be ciphered in more than one way without making it impossible to decipher it subsequently, how could effective cryptanalysis be carried out? The question would remain unanswered for more than 300 years.

Classifying alphabets

Although it took almost eight centuries, the polyalphabetic ciphers such as De Vigenère's square finally outwitted frequency analysis. Curiously, monoalphabetic

systems, despite their weaknesses, had the advantage of being very simple to implement. Cryptographers devoted themselves to refining the procedures and to filling their algorithms with tricks, but fundamentally they kept on using the same concepts as the simplest ciphers.

One of the most successful variants of the monoalphabetic system was that known as the *homophonic* substitution cipher, which attempted to frustrate potential attacks using statistical cryptanalysis by increasing the substitution rates of the letters with the greatest frequency of appearance. So, if the letter E represented, on average, 10 per cent of a text in any language, a homophonic substitution cipher attempted to alter the frequency by replacing the E with 10 alternative characters. Such methods were remained in favour until well into the 18th century.

THE CRYPTOGRAPHERS OF THE SUN KING

Although few outside the court of Louis XIV knew of their existence, the brothers Antoine and Bonaventure Rossignol were two of the most feared men in Europe during the [illegible] of the 17th century. Their ability to decipher messages of the enemies of France (and of the personal enemies of the monarch) was matched by their inventiveness as cryptographers. They developed the *Grande Chiffre* (Great Cipher), a complex algorithm of syllable substitution used to encrypt the king's most important messages. When the brothers died, however, the cipher fell out of use and became unbreakable. Not until 1890 did a cryptography expert, the retired soldier Étienne Bazeries, take on the arduous task of decrypting the ciphered documents and, following years of hard work, became the unsuspecting recipient of the Sun King's secret messages.

Louis XIV in a portrait by Mignard

Things were to move on, though. The emergence of the great nation states and their accompanying diplomacy generated a marked increase in the demand for secure communication. This tendency was further reinforced by the appearance of new communication technologies such as the telegraph, which increased the volume of communications massively. The European states established so-called "black rooms", nerve centres of activity from which the [illegible] were coded and where enemy intercepts were deciphered. The experts working in the black rooms soon made any form of monoalphabetic substitution insecure, however modified it might be. Little by little the great players in the game of information exchange were opting for polyalphabetic algorithms. Having lost their most powerful weapon, frequency analysis, the cryptanalysts were once again left defenceless in the face of the cryptographers' onslaught.

The anonymous cryptanalyst

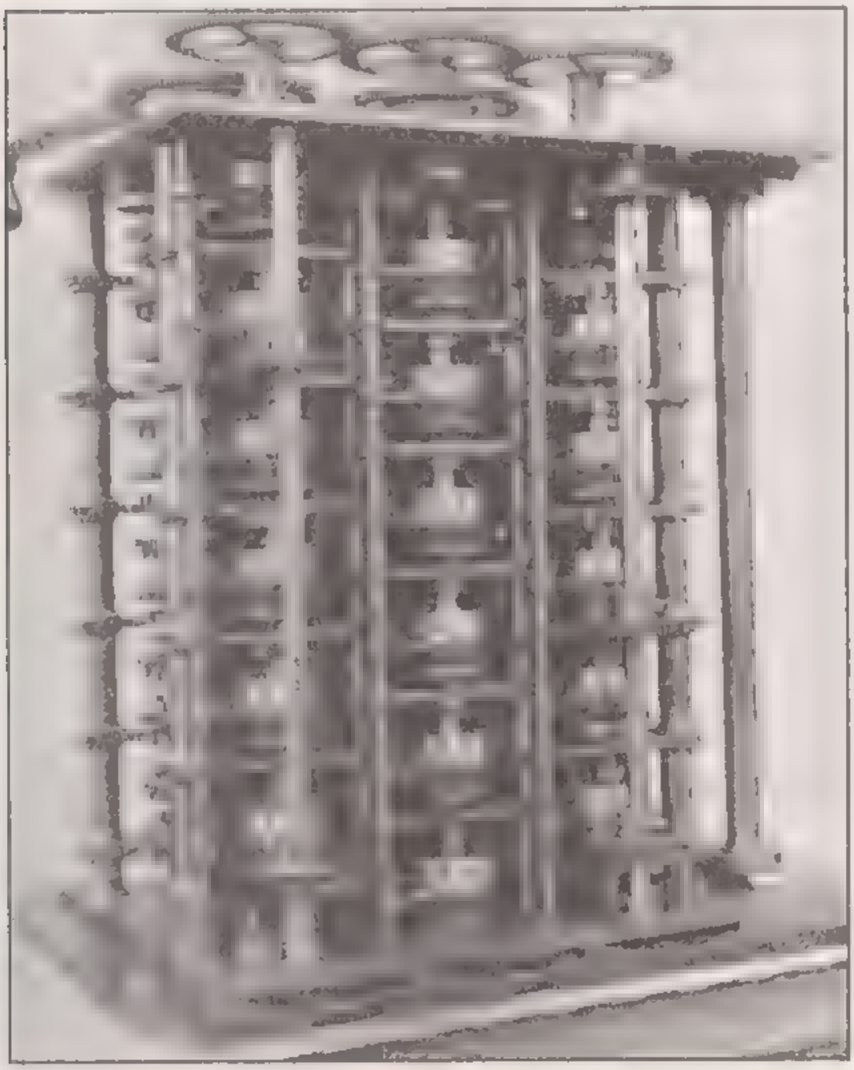
The British mathematician Charles Babbage (1791–1871) was one of the most extraordinary scientific figures of the 19th century. He invented an early mechanical computer called the difference engine that was way ahead of its time, and his interests spanned all the mathematics and technology of the age. Babbage decided to apply his intellect to deciphering polyalphabetic algorithms, with De Vigenère's square (see pages 44 and 47) as his prime target. He focused his attention on one characteristic of this cipher. We should recall that, in the case of De Vigenère's cipher, the length of the chosen keyword determined the number of ciphered alphabets in use. So, if the keyword were "WALK," each letter of the original message could be ciphered in up to 4 different ways. The same would be true of the words. This characteristic would be the toehold from which Babbage would begin to climb the wall of the polyalphabetic cipher. Let's look at the following example of a message ciphered with De Vigenère's square.

Original message	B	Y	L	A	N	D	O	R	B	Y	S	E	A
Keyword	W	A	L	K	W	A	L	K	W	A	L	K	W
Ciphered message	X	Y	W	K	J	D	Z	B	X	Y	D	O	W

What immediately draws our attention is that the word "BY" of the original message is ciphered with the same letters in both cases, XY. This is due to the fact

that the second BY occurs after eight characters and eight is a multiple of the number of letters (four) in the keyword (WALK). With this information, and given a sufficiently long original text, it is possible to guess the length of the keyword. The procedure is as follows: you list all the repeated characters and note after how many characters they repeat. Then you seek whole divisors of these latter numbers. The common divisors are the numbers that are candidates to represent the length of the keyword.

Let's assume that the most probable candidate is 5 because that is the common divisor that appears most often. Now we have to guess what letters each of the five letters of the keyword correspond to. If we recall the encryption process, each letter of the keyword in De Vigenère's square establishes a monoalphabetic cipher of the corresponding letter in the original message. In the case of our hypothetical five letter keyword (C1, C2, C3, C4, C5), the sixth letter (C6) is ciphered with the same alphabet with which the first letter (C1) was ciphered, the seventh (C7)



A working section of Babbage's difference engine, built in 1991 according to the plans left by its inventor. The device allows the approximation of logarithmic and trigonometric functions and, therefore, the calculations of astronomical tables. Babbage did not see it built in his lifetime.

with that used to cipher the second (C2), etc. Therefore, what the cryptanalyst is actually dealing with is five separate monoalphabetic ciphers, each one of which is vulnerable to traditional cryptanalysis.

The process is concluded by designing a frequency table for each of the letters in the ciphered text with the same letters as the keyword (C1, C6, C11...) and C2, C7, C12... until you have the five groups of letters that make up the total length of the message. Then compare these tables with a frequency table of the language of the plaintext message in order to decipher the keyword. If the two data sets do not appear to coincide, we start again with the second most probable length of keyword. This time we identify at least one probable keyword, so all that is left to do is decipher the message. By this method, the polyalphabetic code was broken.

Babbage's astounding exercise, completed around 1854, would, nonetheless, remain in obscurity. The eccentric British intellectual never published his discovery and only recent reviews of his notes have led us to identify him as the pioneer of deciphering polyalphabetic keywords. Fortunately for cryptanalysts the whole world over, a few years later, in 1863, the Prussian officer Friedrich Kasiski published a similar method.

Irrespective of who was the first to break it, the polyalphabetic cipher had ceased to be impregnable. From this moment on, the strength of a cipher was going to depend less on great algorithmic innovations of encryption and more on increasing the number of potential ciphered alphabets, which would have to be so large as to make frequency analysis and its variants completely unfeasible. A parallel objective was to find ways of speeding up cryptanalysis. Both fields of enquiry converged toward the same point and gave birth to the same process: computerisation.

Chapter 3

Coding machines

The 19th century would expand the usefulness of codes way beyond setting secret messages. The development of the telegraph in the first third of the century and, thirty years later, the development of the two way telegraph by Thomas Alva Edison, revolutionised communications and, consequently, the world. Since the telegraph functioned by electrical impulses, it was necessary to implement a system that would translate the content of the messages to a language that a machine could express – and transmit. In other words, a code was needed. From among the various proposals, a system of dots and dashes invented by the American artist and inventor Samuel F. B. Morse prevailed. Morse code can be considered a predecessor of the codes that, many decades later, are used indirectly by us all to enter data into computers and get information back out of them.

Morse code

Morse code represents the letters of the alphabet, numbers and other signs by a combination of dots, dashes and spaces. In this way, it translates the alphabet into something that can be expressed by means of simple signals of light, sound or electricity. Each dot represents a single time unit of approximately 1/25th of a second, a dash is three units long (equivalent to three dots). The spaces between the letters are also three units long, and five units are used as the spaces between words.

At first, Morse was denied a patent on his code in the United States and in Europe. Finally, in 1843, he obtained government financing for the construction of a telegraph line between Washington DC and Baltimore. In 1844, the first coded transmission was performed, and shortly after a company was formed with the express purpose of covering the whole of North America with telegraph lines. By 1860, when Napoleon III awarded Morse the Legion of Honour, the United States and Europe were already criss-crossed by his telegraph wires. At Morse's death in 1872, America had more than 300,000 kilometres of cable.

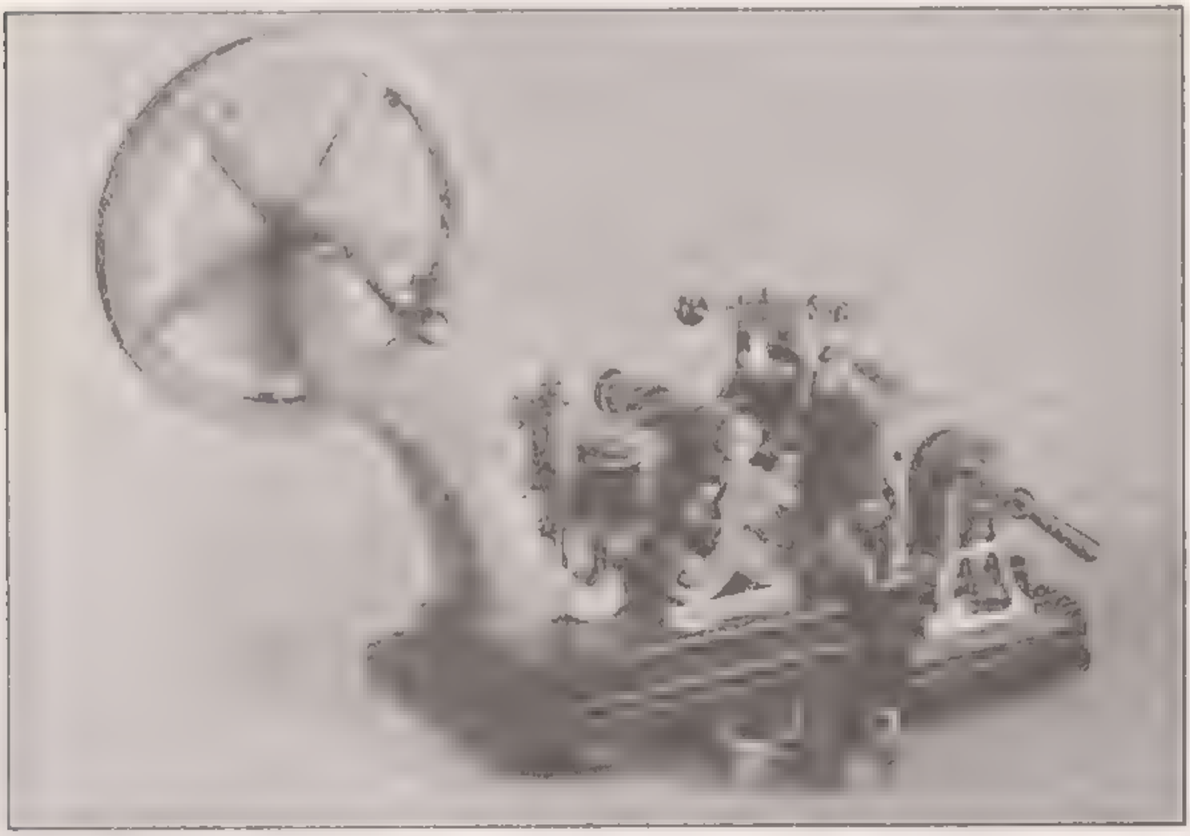
At first, a simple device, invented in 1844 by Morse himself, was used to send and receive telegraph messages. The device consisted of a telegraph key that served

NON-VERBAL COMMUNICATION

[illegible]

him the dialogue of the actors

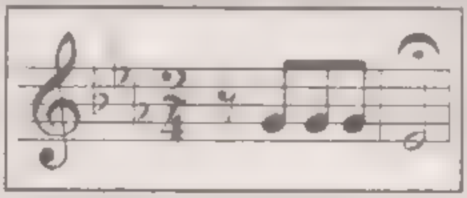
to connect and disconnect the electric current, and an electromagnet that received the incoming signals. Every time the key was pressed down – generally with the index or middle fingers – an electrical contact was established. Intermittent impulses produced by tapping the telegraphic key were transmitted to a cable composed of two copper wires. These wires, supported by tall wooden "telegraph" poles, connected the various different telegraph stations and often extended hundreds of kilometres without interruption.



First telegraph machine designed by Samuel Morse in 1844

SYMPHONY IN V MAJOR

Beethoven is another famous [illegible] only rather indirectly, the first [illegible] rhythm reminiscent of a message in Morse code: "dot dot dot dash."



In Morse code "dot dot dot dash" corresponds to the letter [illegible]
Because of this, the BBC [illegible] Beethoven's *Fifth* as the opening theme for its broadcasts [illegible] cupied Europe during the World War II.

The receiver contained an electromagnet, formed from a coil of copper wire wrapped around an iron core. When the coil received the impulses of the electric current that corresponded to the dots and dashes, the iron core became magnetised and attracted a moving part, also made of iron That produced a distinctive sound when striking the magnet. This sound was a short "click" when a dot was received, and a longer note when a dash was received. Initially, sending a telegram with such a device required a human operator to tap out the codified version of the message at one end, and someone else to receive and decipher it at the other.

The translation of the conventional characters of Morse code was done according the following table:

SIGN	CODE	SIGN	CODE	SIGN	CODE	SIGN	CODE	SIGN	CODE	SIGN	CODE
A	[illegible]	G	[illegible]	N	[illegible]	U	[illegible]	0	[illegible]	[illegible]	[illegible]
B	[illegible]	H	[illegible]	O	– – – ·	V	[illegible]	1	[illegible]	8	[illegible]
C	– · – ·	I	· ·	P	· – – ·	W	· – –	2	· – – –	9	– – – – ·
CH	– – – –	J	· – – –	Q	– – · –	X	– · · –	3	· · · – –	.	· – · – · –
D	[illegible]	K	[illegible]	R	[illegible]	Y	[illegible]	4	[illegible]	,	[illegible]
E	[illegible]	L	[illegible]	S	[illegible]	Z	[illegible]	5	[illegible]	?	[illegible]
F	[illegible]	M	[illegible]	T	[illegible]			6	[illegible]		[illegible]

So the message "I love you" would be coded as:

[illegible]		L	O	V	E		Y	O	U
[illegible]		[illegible]	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]

As mentioned before, Morse code was, in a way, the first version of future digital communication systems. To demonstrate this idea, we could happily convert Morse into numbers, assigning a 1 to the dot and a 0 to the dash. Such strings of 1 and 0 will become more familiar in later chapters.

In the 20th century, traditional telegraphy was replaced by wireless communication driven by the invention of the radio. The telegraphists of yesteryear became radio operators. This new technology meant messages could be sent at even higher speeds and in bulk. However, messages sent as electromagnetic waves were relatively easy to intercept. This provided cryptanalysts with large quantities of ciphered material to work on and helped to consolidate their dominant position in the battle with cryptographers, given that the majority of ciphers used by governments and private agencies, even the most sensitive, were based on known algorithms. This was the case of the Playfair cipher for example, which was invented by the Britons Baron Lyon Playfair and Sir Charles Wheatstone. The Playfair cipher was an ingenious variation on Polybius' cipher, but in the end only a variation – the cipher is set out in detail in the Appendix.

Despite the considerable inventiveness of their creators, the decryption of these recycled ciphers was ultimately a question of time and computing capacity. The cryptographic history of World War I illustrates this perfectly. We have already heard

SAVE OUR SOULS, SHIP OR ANYTHING ELSE BEGINNING WITH 'S'

[illegible] famous signal in Morse Code is SOS. It was established as a distress call by a group of [illegible] countries because of the simplicity of its transmission: three dots, three dashes, three [illegible] was attached to it. However, people were soon giving the signal alternative [illegible] of these "backronyms" was Save Our Souls [illegible] as the signal was [illegible] SOS [illegible] became referred to popularly as Save Our [illegible]

about the weakness of the German diplomatic cipher during the Zimmermann telegram incident. What the Germans themselves didn't suspect was that another of their common ciphers, known as ADFGVX and used to encrypt the most sensitive messages destined for the front, could also be solved by enemy cryptanalysts despite its supposed invulnerability. This double failure of Germany's World War I codes made all sides aware of the need to cipher more securely. This objective was to be achieved by making cryptanalysis more difficult.

80 kilometres from Paris

In June, 1918, German troops were preparing to attack the French capital. It was essential to the Allies to intercept enemy communications to find out where the offensive incursions would take place. The German messages destined for the front were encrypted with the ADFGVX cipher, considered by the German military to be unbreakable.

Our interest in this cipher stems from the fact that it combines substitution and transposition algorithms. It is one of the most sophisticated methods of classical cryptography. Introduced by the Germans in March 1918, no sooner did the French learn of its existence than they frantically applied themselves to breaking the code. Luckily for them, a talented cryptanalyst called Georges Painvin was working in the central cipher bureau. He devoted himself to the task day and night. The night of June 2, 1918, Painvin succeeded in deciphering a first message. The ominous content was an order directed to the front: "Rush munitions. Even by day if not seen." The introduction to the cipher indicated that it had been sent from some place located between Montdidier and Compiegne, some 80 kilometres north of Paris. Painvin's achievement allowed the French to foil the attack and halt the German advance.

As mentioned already, the ADFGVX cipher consists of two parts: a substitution and a transposition. In the first phase – substitution – we have a seven-by-seven grid in which the first row and the first column each contain the letters ADFGVX (see page 58). The remaining squares of the grid are randomly filled in with 36 characters: the 26 letters of the alphabet and the numbers 0 to 9. The arrangement of the characters constitutes the key to the cipher, and the recipient, clearly, needs this information to understand the content of the message.

Let's use the following base table:

	A	D	F	G	V	X
A	O	P	F	C	Z	C
D	G	3	B	H	4	K
F	A	1	7	J	R	2
G	5	6	.	D	E	T
V	V	M	S	N	Q	I
X	U	W	9	X	Y	8

The cipher consists of translating each character of the message into coordinates using the letters from the group ADFGVX. The first coordinate is the letter that corresponds to the row, and in the second one corresponds to the column. For example, if we wished to cipher the number 4, we would write "DV." The message "Target is Paris" would be ciphered as follows:

T	a	r	g	e	t		s	P	a	r	i	s
GX	FA	FV	DA	GV	GX	VX	VF	AD	FA	FV	VX	VF

Up to this point we are dealing with a simple substitution, and frequency analysis would be sufficient to decipher the message.

The cipher, however, contains a second phase – transposition. The transposition depends on a keyword agreed upon by the sender and the receiver This phase of the cipher is carried out as follows. First, we construct a grid with as many columns as there are letters in the keyword, and we fill in the cells with the ciphered text. The letters of the keyword are written in the top row of the new grid In this example, the keyword will be BETA. We create a new table in which the first row consists of the keyword and the following rows contain the letters obtained by encoding the message through substitution Any empty cells are filled in with the number zero which, as we see from the first table, is symbolised by AG.

So to apply this second process to our message "Target is Paris", we first recall that the substitution cipher produced was:

GX	FA	FV	DA	GV	GX	VX	VF	AD	FA	[illegible]	[illegible]	VF

When we apply BETA as the keyword, a new table ensues

B	E	T	A
G	X	F	A
F	V	D	A
G	V	G	X
V	X	V	F
A	D	F	A
F	V	V	X
V	F	A	G

We continue with the transposition cipher and change the position of the columns, so the letters of the key are arranged in alphabetical order. This gives us the following table.

A	B	E	T
A	G	X	F
A	F	V	D
X	G	V	G
F	V	X	V
A	A	D	F
X	F	V	V
G	V	F	A

The encrypted message is produced by taking the letters of the grid by columns. In the example, we get:

AAXFAXGGFGVAFVXVVXDVFFDGVFVA

As we can see, the message consists of an apparently random mix of the letters A, D, F, G, V and X. The Germans selected these six letters because they sounded very different to each other when sent in Morse code. This helped the receiver to detect hypothetical transmission errors more easily. Moreover, since it consisted of only six letters, the telegraphic transmission was simple and therefore easy for inexperienced operators to send.

If we turn to the Morse code table at the beginning of the chapter, we can see that the codes for each of the letters of the cipher ADFGVX are as follows:

A
D
F
G
V
X

The receiver only needs the random distribution of the letters and numbers shown by the first table and the second keyword to reverse the encryption and reveal the message.

The Enigma machine

In 1918 the German engineer Arthur Scherbius patented a machine that was designed to produce completely secure communications. Its name, Enigma, has since become synonymous with military secrecy. For all its apparent sophistication, Enigma was, in essence, an improved version of Alberti's disk, as we shall see below.

Because it was relatively easy to use and because of the complexity of the resulting cipher, Enigma was the system selected by the German government to encrypt a large part of its military communications [illegible].

As a result, deciphering the Enigma code [illegible] priority for the governments confronting Nazi Germany. W[illegible] intercepted and deciphered by Allied intelligence [illegible] ing an end to the conflict. The history of the deciphering [illegible] fascinating story that involved, in the main, the departments of [illegible] Poland and the United Kingdom, and includes among its heroes th[illegible] genius Alan Turing, the man considered to be the father of modern [illegible]. The battle to break the Enigma code also yielded the first digital computer [illegible] tory, and can be considered the most spectacular episode in the long and [illegible] history of military cryptanalysis.

Above left: German soldiers transcribe a ciphered message [illegible] Enigma [illegible] during World War II. Above right: a replica four-rotored Enigma machine

The Enigma machine itself was an electromagnetic device similar in appearance to a typewriter. What made it so special was that its mechanical components changed position with each key press so that even if the same plaintext letter was pressed consecutively, it would most likely be encoded differently each time.

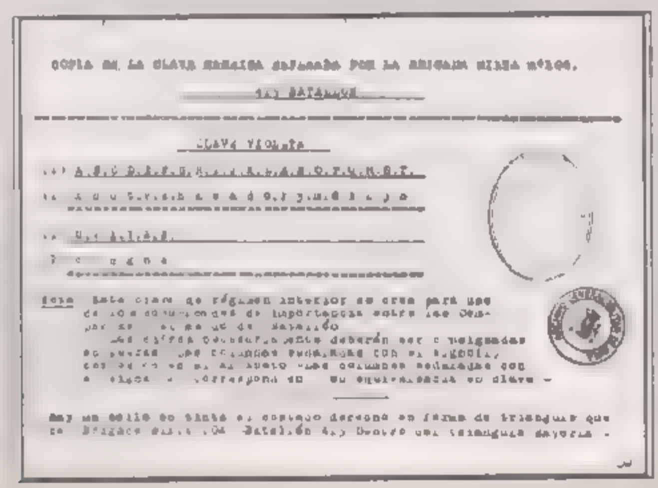
The physical process of ciphering was relatively simple. First, the sender arranged the machine's various plugs and rotors according to a starting point specified by the particular code book in force at the time (code books were changed regularly). Then he would type the first letter of the plaintext, and the machine would automatically generate an alternative letter that would appear on an illuminated panel – the first letter of the ciphered message.

TRENCH CODES

In battle, using complex ciphers like ADFGVX is very hard work. In the Spanish Civil War (1936–1939), for example, there were many simpler substitution algorithms, such as the following:

A	B	C	D	E	F	G	H	Y	J
53 91	12 70	41 86	31	27 43	24	16	11	40,59	22
L	M	N	O	P	Q	S	R	T	U
13	15	96 66	84,39	75	71	28,54	28 54	19	74 44

As we can see, several letters have more than one ciphered version. The R, for example, can be substituted by 28 or by 54. The word 'GUERRA' (WAR) would be ciphered as 167427285453. These codes, which were primarily substitution codes, were called trench codes and were intended for very specific uses.

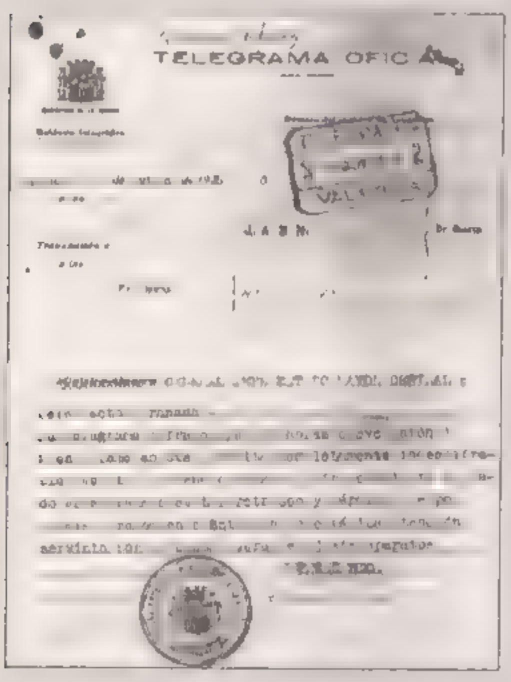


The Clave Violeta (Violet Key, left) was used by the 415th battalion of the 104th Republican Brigade, and was captured by the Nationalist side. The note translates as: "The ciphers will necessarily have to be represented as letters. The columns [rows] marked with a (1) correspond to the alphabet. The columns marked with a (2) correspond to their equivalent in code."

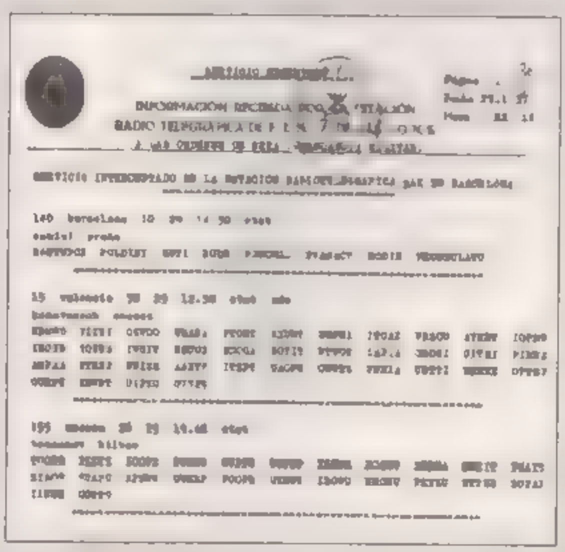
The first rotor switch made a rotation that placed it in one of the 26 possible positions. The switch's new position brought a new cipher of the letters, and the signals operator then entered the second letter, and so on. To decode the message, it was sufficient to enter the ciphered characters into another Enigma machine as long as the starting parameters of the second machine were the same as those of the machine that had carried out the encryption.

Using the illustration on the following page, we can present a very simplified schematic of the Enigma's encryption mechanism, using rotors with an alphabet of only three letters. As a result each rotor has only three possible positions instead of the 26 in the real thing.

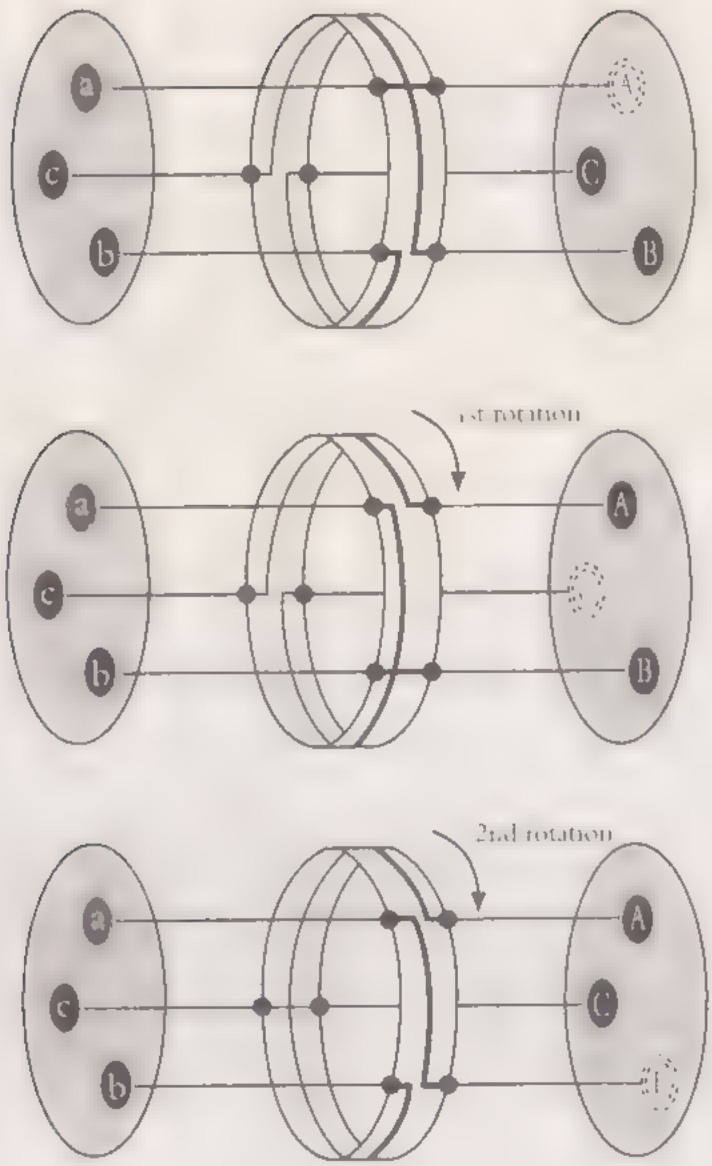
For a higher level of secrecy, the Nationalist side, headed by General Franco, deployed another weapon: 30 of the so-called Enigma machines supplied by their Nazi allies. This would be the first intensive military use of the ciphering device that Germany would come to use in World War II. The British attempted to break the code during the Spanish conflict, but without success.



Telegram (left) of October 27, 1936, to the chief of the Granada Sector (Republican): "Your telegram ciphered yesterday...proved indecipherable."



An encoded Republican message (right) intercepted by the Spanish Falangist Fascist movement in the Canary Islands



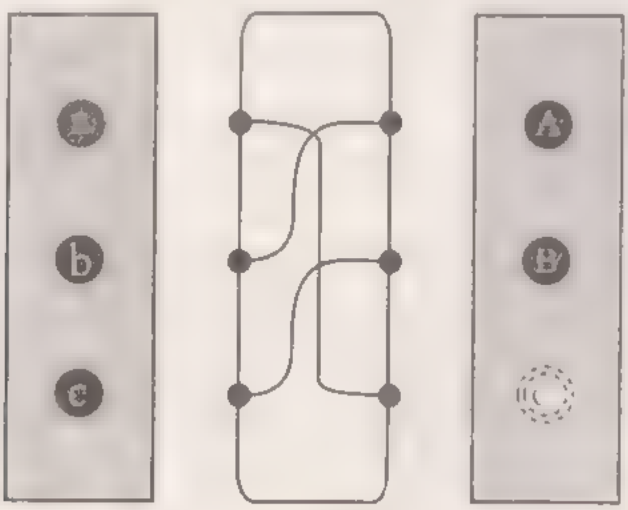
As we can see, with an Enigma machine's rotor in the initial position, each letter of the original message is substituted by a different one except for A, which remains unchanged. After ciphering the first letter, the rotor does a one-third turn. In this new position, the letters are now substituted by different ones from those of the first cipher. The process concludes with the third letter, after which the rotor returns to its initial position and the sequence of the cipher will repeat itself.

The rotary switches of a standard Enigma machine had 26 positions, one for each letter of the alphabet. Consequently, a single rotor could perform 26 different ciphers. Therefore, the initial position of the rotor is the key. To increase the number of possible keys, the design of the Enigma incorporated up to three rotors, connected mechanically one to the other.

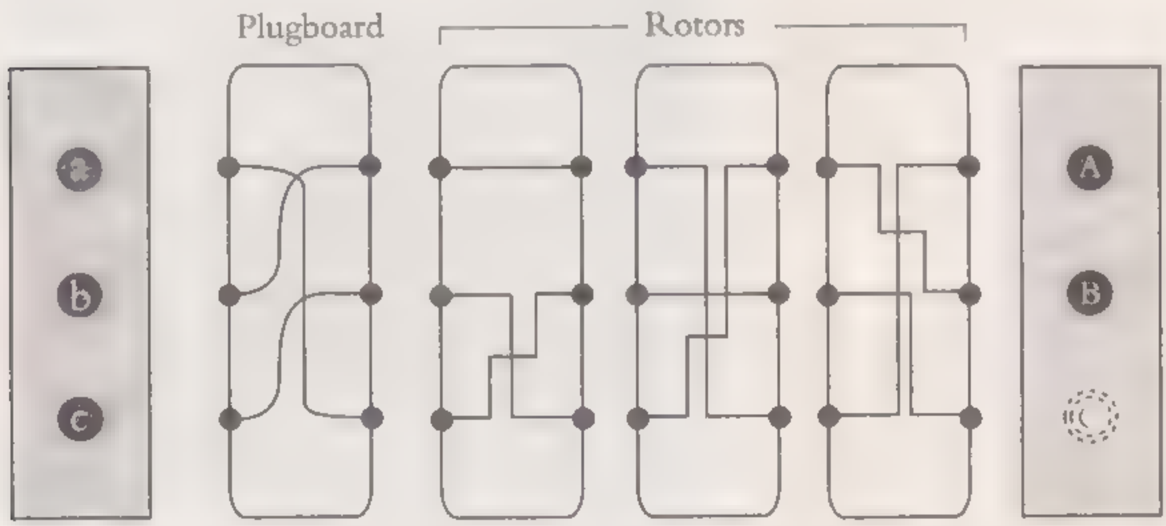
So, when the first rotor completed a turn, the next one initiated another one, and so on until the complete rotations of all the rotors ended, for a total of $26 \times 26 \times 26 = 17{,}576$ possible ciphers. In addition, Scherbius's design allowed for exchanging the order of the switches, thus increasing the number of codes even more, as we shall see below.

A three rotored Enigma machine with its casing partly removed to show its plugboard (at the front).

Besides the three rotors, Enigma also had a plugboard located between the first rotor and the keyboard. The plugboard allowed for the interchange of pairs of letters before they were connected to the switch, and in this way added a considerable number of codes to the cipher. The standard design of the Enigma machine had six cables that could interchange up to six pairs of letters. The following illustration shows the operation of the interchanging plugboard again in a simplified form of only three letters and three cables.



In this way, the A swaps with the C, the B with the A, and the C with the B. With the addition of a plugboard, a simplified three-letter Enigma machine would function as follows:



How many more codes did the seemingly trivial addition of the plugboard provide? We have to consider the number of ways of connecting the six pairs of letters selected from a group of 26. The possible number of transformations of n pairs of letters of an alphabet of N characters is determined by the following formula:

$$\frac{N!}{(N-2n)!\,n!\,2^n}$$

In our example, $N = 26$ and $n = 6$, and that gives us a mere 100,391,791,500 combinations.

Consequently, the total number of ciphers offered by the Enigma machine with three 26-letter rotors and a plugboard with six cables is the following:

1. With reference to the rotations of the rotary switches, $26^3 = 26 \cdot 26 \cdot 26 = 17{,}576$ combinations.
2. Likewise, the three rotors (1, 2, 3) could interchange with each other and could occupy the positions 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2, 3-2-1; this gives us six possible additional combinations.
3. Finally, we have calculated that the arrangement of the six cables of the initial plugboard added 100,391,791,500 additional ciphers.

The total number of ciphers is obtained from the product of the different specified combinations, [illegible] Therefore, Enigma machines could [illegible] million different combinations. [illegible] that their highest level communi[illegible]

Deciphering the Enigma code

Any Enigma key first specified the configuration of the [illegible] six possible letter interchanges – for example, B/Z, F/Y, K/[illegible] which indicated that the first cable interchanged the letters B and Z [illegible] Secondly the key showed the order of the rotors (such as 2 [illegible]) and [illegible] included the starting orientation of rotors (such as R, V, B) indicating which letter was located at the starting point, or index mark. These settings were [illegible] in code books that were themselves transmitted in an encrypted form and could change from one day to the next or when other circumstances dictated. For example, certain keys were reserved for certain types of message.

To avoid repeating the same code throughout the day – during which thousands of messages could be sent – Enigma's operators had some [illegible] for transmitting new codes, of restricted use, without having to alter the entire book of shared codes. So, the despatcher sent a six-letter message [illegible] the applicable daily code, that was actually a new [illegible] for example FYJ. (For greater security the [illegible] twice, hence the six letters.) Next, he would code the [illegible] new arrangement. The recipient received a message [illegible] the code of the day but he knew that the first six letters were [illegible] to arrange the rotors in another position. The receiver would do this keeping the plugboard and the order of the rotors unchanged [illegible] the message.

The Allies obtained the first valuable information relating to Enigma in 1931 from a German spy, Hans Thilo Schmidt. This consisted of various [illegible] for the practical use of the machine. The contact with Schmidt was made by French intelligence services who subsequently shared information with their Polish counterparts. The Polish department of cryptanalysis, the *Biuro Szyfrów* [illegible] went to work on Schmidt's documents and it got hold of various Enigma machines stolen from the Germans.

In an unusual move for the time, the Polish code-breaking team included a large number of mathematicians. Among them was a talented, introspective and shy young man of 23 by the name of Marian Rejewski. He immediately concentrated his efforts on the six-letter codes that preceded many of the daily messages exchanged by the Germans. Rejewski theorised that the second three letters of the code were a new cipher of the first three and knew, therefore, the fourth, fifth and sixth letters could give a clue to the rotation of the switches.

From this discovery, as small as it might appear, Rejewski built an extraordinary network of deductions that would lead to the breaking of the Enigma code. The details of this process are very complex, and we will not expound them here, but the fact is that, after a few months, Rejewski had reduced the number of possible codes that needed to be deciphered from ten thousand billion to just 105,456 that resulted from different combinations of the order of the switches and their different rotations. To do this, Rejewski built a device, known as the Bombe, that functioned in the same way as the Enigma and that could simulate any of the possible positions of the three rotors in search of the daily code. As early as 1934, the Biuro Szyfrów had broken Enigma and could decipher any message within 24 hours.

Although the Germans did not know that the Poles had penetrated Enigma's security, they still added improvements to a system that, after all, had already been operating for more than a decade. In 1938, the Enigma operators received two more rotors to add to the three standard positions and, shortly thereafter, new models of the machine were distributed with ten cable pegboards.

Suddenly, the number of possible codes increased to about 159 quintillion. The addition, alone, of two more rotors to the rotation of the switches increased the possible combination of arrangements from six to 60. That is, any one of the five rotors in the first position (five options) multiplied by any one of the four remaining rotors in the second position (four options) multiplied by any one of the three rotors in the third position (three options) = 5 x 4 x 3 = 60. Although they knew how to decipher the code, the Biuro Szyfrów lacked the means necessary to analyse 10 times as many new rotor configurations all at once.

Some versions of the Enigma machine

The British take over

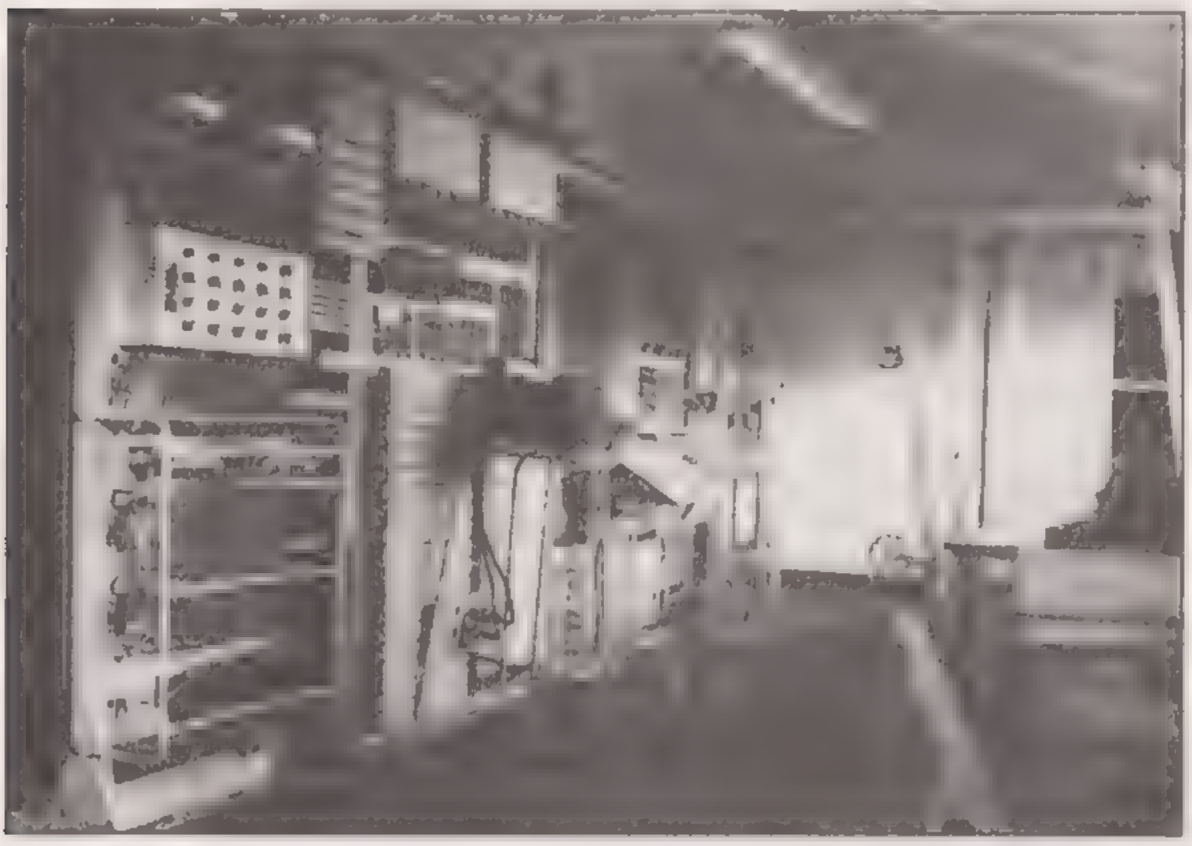
The upgrade to the Enigma system was not accidental. Germany had already begun its aggressive expansion through Europe with the annexation of Czechoslovakia and Austria, and was planning the invasion of Poland. In 1939, with the conflict now unleashed in the heart of Europe, and their country conquered, the Poles transferred all their Enigma machines and understanding to their British allies who, in August of that year, decided to bring together their previously dispersed cryptanalytic units. The location selected was a mansion situated on the outskirts of London, in an estate called Bletchley Park. A brilliant new cryptanalyst was added to the team at Bletchley Park, a young Cambridge mathematician called Alan Turing. Turing was a world authority in the sphere of computing, then still an embryonic field, and open to new and revolutionary developments. Deciphering the improved Enigma machines proved to be the impetus behind several leaps forward in computing.

Experts at work at Bletchley Park where the Enigma code was deciphered.

The experts at Bletchley Park concentrated on short fragments of ciphered text that they suspected corresponded to segments of plaintext. For example, thanks to their spies on the ground, it was known that the Germans had the habit of transmitting a codified message about the meteorological conditions at various locations along the front line around 6 p.m. every day. Therefore, they were reasonably certain that a message intercepted shortly after that hour contained a ciphered version of plaintexts such as "weather" and "rain." Turing invented an electrical system that allowed for the reproduction of all and every one of the 1,054,650 possible combinations of the order and position of the three rotors in less than five hours. This system was fed with ciphered words that, by the length of their characters and other clues, were suspected to correspond to fragments of plaintext such as the above mentioned weather and rain.

Let us suppose that they suspected that the text ciphered FGRTY was an encrypted version of "bread". The cipher would be entered into the machine and if there was a combination of rotors that gave the word "bread" as a result, the cryptographers knew that they had found the codes that corresponded to the configuration [illegible] switches. Next, the operator entered the ciphered text in a real Enigma [illegible] with the rotors arranged according to the code. If the machine showed a plaintext DREAB, for example, it was clear that the part of the code relating

to the position of the plugboard cables included the transposition of the letters D and B. In this way, they obtained the entire code. Enigma's secrets were definitely becoming known. In the process of developing and refining the above-mentioned analytic mechanisms, the team at Bletchley Park built the first digital and programmable computer in history, christened Colossus.



Colossus, the forerunner of the modern computer, at Bletchley Park. The photograph, taken in 1943, shows the control panel of the complex device

Other ciphers of World War II

Japan developed two of its own encoding systems known as Purple and JN-25. The first one was used for diplomatic communications and the second to send military messages. Both ciphers were carried out by mechanical devices. JN-25, for example, consisted of a substitution algorithm that translated the written characters of the Japanese language (up to a limit of 30,000 characters) into series of numbers as specified by random tables of five number groups. Despite the precautions taken by the Japanese, the British and Americans cracked the Purple and the JN-25 codes. The intelligence obtained thanks to the interception of the Purple and

JN-25 ciphers was codenamed Magic, and had considerable impact during pivotal encounters in the Pacific war, particularly the Battles of the Coral Sea and Midway, both in 1942. Magic's intelligence was also used to plan strategic missions, such as the interception and shooting down of Japanese military commander Admiral Yamamoto's plane the following year.

A TRULY BRILLIANT MIND

Alan Turing (left) was born in England in 1912. Even when young, he showed a great aptitude for mathematics and physics. In 1931, he went to Cambridge University where he became interested in the work of the logician Kurt Gödel into the general problem of inherent incompleteness of any logical system. Three years before he had published a study on the theoretical possibility of building machines that were capable of computing different algorithms such as addition, multiplication, etc. Inspired by Gödel's works, in 1937 Turing took his ideas on the limits of proof and computation a step forward and established the principles of a 'universal machine' capable of performing any conceivable algorithmic computation. Thus was born one of the pillars of modern information theory. Two years before, Turing had made contact with the great Hungarian mathematician János von Neumann, who was by that time living in the United States and better known as John von Neumann. Considered the 'other father' of computing, he offered Turing a job at Princeton, a well-paid and highly prestigious position. However, Turing preferred the bohemian atmosphere at Cambridge and declined the offer. In 1939, as war broke out, he joined the British cryptanalysts [illegible] at Bletchley Park. His work during the war earned him an OBE (Order of the British [illegible]). [illegible] Turing was a homosexual (illegal at the time) and a conviction in 1952 made it [illegible] to work on secret government projects. [illegible], depressed by the rejection, Turing committed suicide on June 8, 1954 by swallowing potassium cyanide.

The Navajo code talkers

While the United States made good use of information intercepted from the enemy in the Pacific theatre of operations, the US military's own communications used several codes – in the strict sense of the word as discussed at the beginning of the book. The encryption algorithms operated directly on the nature of the words. These codes – the Choctaw, the Comanche, the Meskwaki and above all, the Navajo – were not explicitly set out in complicated manuals, nor were they the result of planning by a judicious department of cryptographers: they were simply authentic Native American languages.

The United States army placed radio operators from these native groups in various units along the front, and charged them with transmitting messages in their respective languages, which were unknown not only to the Japanese, but also to the rest of the American forces. A set of basic codes was superimposed on these ciphered messages to prevent a captured soldier from being forced to translate them. These "code talkers" served in American units until the Korean War.

Two Navajo "code talkers" during the Battle of Bougainville in 1943

Innovations: Hill's cipher

The ciphers discussed up to this point, in which one character is substituted by another in some pre-established manner, are always vulnerable to being cracked by cryptanalysis, as we have seen.

In 1929, the US mathematician Lester S. Hill invented, patented and put up for sale – unsuccessfully – a new ciphering system that made use of a combination of modular arithmetic and linear algebra.

As we shall see below, a matrix can be a very useful tool to cipher a message, by composing the text into pairs of letters and associating each letter with a numerical value.

To cipher a message, we use a matrix:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

with the restriction that its *determinant* be 1, that is, that $ad\text{-}bc = 1$. To decipher it, we use the inverse matrix:

$$A^{-1} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

A BRUSHSTROKE OF LINEAR ALGEBRA

A matrix can be defined as a table arranged firstly in rows and then columns. For example, a matrix of 2 x 2 takes the form

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

and a matrix of 2 x 1 is of the form: $\begin{pmatrix} x \\ y \end{pmatrix}$.

The product of both these matrices gives us a new matrix 2 x 1, called a *column vector*:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}$$

In the case of the matrix 2 x 2, the value *ad-bc* is called the determinant of the matrix.

The restriction in the value of the determinant is set so that the inverse matrix will function as a deciphering tool. As a rule, for an alphabet of n characters, it is necessary that the gcd (the determinant of A, n) = 1. If the opposite were true, the existence of the inverse in modular arithmetic could not be guaranteed.

Continuing the example, we take an alphabet of 26 letters with a "blank space" character, which for purposes of this example we will designate as α. We assign each letter with a numerical value as shown in the following table.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	α
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

To obtain values between 0 and 26, we will work in modulus 27.

The process of ciphering and deciphering the text is as follows. First we determine a ciphered matrix A with determinant 1.

For example, $A=\begin{pmatrix} 1 & 3 \\ 2 & 7 \end{pmatrix}$.

The deciphered matrix will be the inverse matrix $A^{-1}=\begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix}$.

Therefore, A will be the key of the cipher, and A^{-1} is the decipher key.

Below, for example, we establish the message "BOY." The letters of the message are grouped in pairs: BO Yα. Their numerical equivalents according to the table are the pairs of numbers (1, 14), and (24, 26). Next we multiply matrix A by each pair of numbers:

$$\text{Ciphered "BO"} = \text{BO} = \begin{pmatrix} 1 & 3 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 1 \\ 14 \end{pmatrix} = \begin{pmatrix} 43 \\ 100 \end{pmatrix} \equiv \begin{pmatrix} 16 \\ 19 \end{pmatrix} (\text{mod } 27),$$

that, according to the table, corresponds to the letters (Q, T).

$$\text{Ciphered "Y}\alpha\text{"} = \text{Y}\alpha = \begin{pmatrix} 1 & 3 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} 24 \\ 26 \end{pmatrix} = \begin{pmatrix} 102 \\ 230 \end{pmatrix} \equiv \begin{pmatrix} 21 \\ 14 \end{pmatrix} (\text{mod } 27),$$

that corresponds to the letters (V, O).

The message "BOY" is ciphered "QTVO"

For the deciphering, the inverse operation is performed using the matrix:

$$A^{-1} = \begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix}.$$

We take the pair of letters (Q, T) and seek their numerical equivalents from the table: (16, 19). We then multiply them by A^{-1}, and get:

$$\begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 16 \\ 19 \end{pmatrix} = \begin{pmatrix} 55 \\ -13 \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 14 \end{pmatrix} \text{ (mod. 27), equivalent to (B, O)}$$

We do the same with the second pair (V, O) and their numerical values (21, 14) and we get:

$$\begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 21 \\ 14 \end{pmatrix} = \begin{pmatrix} 105 \\ -28 \end{pmatrix} \equiv \begin{pmatrix} 24 \\ 26 \end{pmatrix} \text{ (mod. 27), equivalent to (Y, @).}$$

We have then proven that the deciphering key works.

For this example we have considered pairs of two characters We would have greater security if we grouped the letters in threes or even fours. In these cases, the calculations would be made with matrices 3 x 3 and 4 x 4, respectively, which would be extremely laborious if carried out manually With today's computers, however, it is possible to work with huge matrices, and with their respective inverses.

Hill's cipher suffers from an important weakness. if the recipient has a small fragment of the plaintext, it is possible to decipher the entire message. The search for the perfect cipher was far from over.

Chapter 4

Communicating With 0 and 1

The invention of the Colossus computer and the breaking of the Enigma code opened the door to the greatest communication revolution known to humanity. This gigantic step forward was based to a large extent on the development of an encryption system that enabled secure, efficient and rapid communications across a vast network driven by two fundamental agents: computers and their users – you and me. When we use the word *security* today, we are not just referring to cryptography and secrecy. The word also has a much broader sense that also encompasses notions of reliability and efficiency.

The binary system forms the basis of the technological revolution. This super simple code formed by two characters, 0 and 1, is used in computing for its ability to represent the interaction of the electronic circuits in a computer (i.e. a circuit is on, represented by 1, or off, represented by 0). Each 0 and each 1 is termed a *bit* (a term derived from *binary digit*).

The ASCII code

One of the binary system's many applications is a specific family of characters each with a length of 8 bits – known as a *byte*. These characters are *alphanumeric* and represent the basic symbols used in conventional communication. They are termed the ASCII (American Standard Code for Information Interchange) codes. The number of ways of arranging 0 and 1 in a group is: $2^8 = 256$.

ASCII codes allows users to enter text into a computer. When we type an alpha

MEMORY BYTES

The memory and storage capacity of a computer is measured in multiples of bytes:

Kilobyte (kB): 1,024 bytes

Megabyte (MB): 1,048,576 bytes

Gigabyte (GB): 1,073,741,824 bytes

Terabyte (TB): 1,099,511,627,776 bytes

numeric character, the computer converts it into a byte of data – a chain of eight bits. So, for example, if we type the letter A, the computer converts it into 0100 0001.

Binary ASCII values are given to all the characters in common usage – 26 capital letters, 26 lower-case letters, 10 numerical digits, 7 symbols of punctuation and some special characters. All are shown in the following table. The corresponding decimal number (in the column headed 'Dec') is given for each character's binary code:

ASCII TABLE								
Character	Binary	Dec	Character	Binary	Dec	Characte	Binary	Dec
(space)	0010 0000	32	@	0100 0000	64		0110 0000	96
!	0010 0001	33	A	0100 0001	65	a	0110 0001	97
"	0010 0010	34	B	0100 0010	66	b	0110 0010	98
#	0010 0011	35	C	0100 0011	67	c	0110 0011	99
$	0010 0100	36	D	0100 0100	68	d	0110 0100	100
%	0010 0101	37	E	0100 0101	69	e	0110 0101	101
&	0010 0110	38	F	0100 0110	70	f	0110 0110	102
	0010 0111	39	G	0100 0111	71	g	0110 0111	103
(	0010 1000	40	H	0100 1000	72	h	0110 1000	104
)	0010 1001	41	I	0100 1001	73	i	0110 1001	105
*	0010 1010	42	J	0100 1010	74	j	0110 1010	106
+	0010 1011	43	K	0100 1011	75	k	0110 1011	107
,	0010 1100	44	L	0100 1100	76	l	0110 1100	108
-	0010 1101	45	M	0100 1101	77	m	0110 1101	109
	0010 1110	46	N	0100 1110	78	n	0110 1110	110
/	0010 1111	47	O	0100 1111	79	o	0110 1111	111
0	0011 0000	48	P	0101 0000	80	p	0111 0000	112
1	0011 0001	49	Q	0101 0001	81	q	0111 0001	113
2	0011 0010	50	R	0101 0010	82	r	0111 0010	114
3	0011 0011	51	S	0101 0011	83	s	0111 0011	115
4	0011 0100	52	T	0101 0100	84	t	0111 0100	116
5	0011 0101	53	U	0101 0101	85	u	0111 0101	117
6	0011 0110	54	V	0101 0110	86	v	0111 0110	118
7	0011 0111	55	W	0101 0111	87	w	0111 0111	119
8	0011 1000	56	X	0101 1000	88	x	0111 1000	120
9	0011 1001	57	Y	0101 1001	89	y	0111 1001	121
	0011 1010	58	Z	0101 1010	90	z	0111 1010	122
	0011 1011	59	[	0101 1011	91	{	0111 1011	123
<	0011 1100	60	\	0101 1100	92	|	0111 1100	124
=	0011 1101	61	]	0101 1101	93	}	0111 1101	125
>	0011 1110	62	^	0101 1110	94	~	0111 1110	126
?	0011 1111	63		0101 1111	95			

When typing "GOTO 2", a phrase in the programming language BASIC, the computer would translate the characters into the corresponding binary sequence:

Typed word	G	O	T	O	Blank space	2
Translation into computer language	01000111	01001111	01010100	01001111	00100000	00110010

The computer would thus execute the sequence:

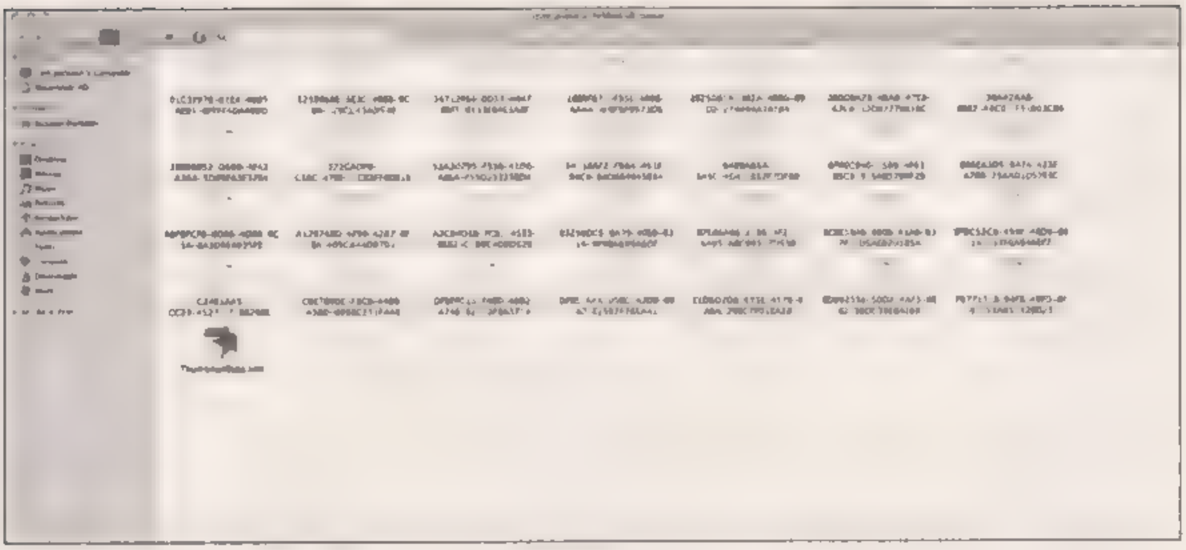
010001110100111101010100010011110010000000110010

The Hexadecimal system

The hexadecimal system is another notable code used in computing. It is a number system that works with sixteen unique digits (hence hexadecimal), as opposed to the normal system that uses ten (decimal). One could say that the hexadecimal system is the computer's second language after binary. Why a 16-digit system? Remember that the computer's basic unit of operation, the byte, is composed of eight bits, which produces up to $2^8 = 256$ different combinations of 0 and 1. $2^8 = 2^4 \times 2^4 = 16 \times 16$. In other words, the combination of two hexadecimal number equals 1 byte.

The sixteen digits of a hexadecimal system are the traditional 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and six more established by convention: A, B, C, D, E, F. To count in a hexadecimal system, we do as follows:

From 0 to 15: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
From 16 to 31: 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F.
From 32 on: 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C...



These files were generated automatically by a computer. Their strange names are actually hexadecimal numbers.

Hexadecimal digits do not distinguish between upper and lower case letters (1E means the same as 1e). The following table shows the first 16 binary numbers and their hexadecimal equivalents:

Binary	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

To go from binary to hexadecimal, we group the bits in four groups of four from the right, and we complete the conversion according to the previous table. If the number of binary digits is not a multiple of four, we fill in the difference with 0 from the left. To go from hexadecimal to binary, we convert each hexadecimal digit into its binary equivalent, as in the following example:

$9F2_{16}$ (is the formal notation of a hexadecimal number, denoted by the subscript 16). Remember the corresponding binary is:

9	F	2
1001	1111	0010

so $9F2_{16} = 100111110010_2$ (Note: the subscript 2 indicates that the number is expressed in a binary system).

Let's now carry out the reverse process. 1110100110_2 has ten digits. Therefore, we complete the number with two zeroes on the left to have 12 digits that we can group by fours.

We convert:

$$1110100110_2 = 0011\ \ 1010\ 0110_2 = 3A6_{16}.$$

What is the relationship between hexadecimal characters and ASCII codes? Every ASCII code contains eight bits (one byte) of information, therefore five ASCII characters contain 40 bits (five bytes) and, since a hexadecimal character contains four bits, we conclude that five ASCII characters are 10 hexadecimal characters.

Let's see an example of coding a phrase in hexadecimal code. Let's try it with the name "NotRealCo Ltd", following these steps.

1. We translate "NotRealCo Ltd" into its binary version with standard ASCII.
2. We group the digits by fours (If the length of the binary string is not a multiple of four, we add 0 to the left).
3. We consult the binary and hexadecimal conversion table and continue with the translation.

Message	N	o	t	R	e	a	l	C	o	
Binary equivalence according to ASCII	01001110	01101111	01110100	01110010	01100101	01100001	01101100	01100011	01101111	00100000
Hexadecimal translation	4E	6F	74	72	65	61	6C	63	6F	20

Message (cont.)	L	t	d
Binary equivalence according to ASCII	01001011	01110100	01100100
Hexadecimal translation	4B	74	64

Therefore, the phrase "NotRealCo Ltd" ciphered in hexadecimal, is as follows.

4E 6F 74 72 65 61 6C 63 6F 20 48 74 64

Numeral systems and base changes

A numeral system of n digits is also said to be of base n. Human hands have ten fingers, and that is probably why the decimal numeral system was invented – counting was carried out with fingers. A decimal number such as 7392 represents a quantity equal to 7 thousands 3 hundreds 9 tens and 2 units. Thousands, hundreds, tens, units are powers of a base number system; in this case, 10. The number 7392, therefore, could be expressed as:

$$7392 = 7 \cdot 10^3 + 3 \cdot 10^2 + 9 \cdot 10^1 + 2 \cdot 10^0.$$

However there is an implicit agreement that we only write the coefficients (7, 3, 9 and 2). Besides the decimal system, there are many other numeral systems (in fact, their total number is infinite). In this volume we have paid special attention to two systems: the binary system of base 2, and the hexadecimal, of base 16. In a binary numeral system, the coefficients only have two possible values: 0 and 1. The digits of the binary numbers are coefficients of the power of 2. So, the number 11011_2 could also be written as

$$11011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0.$$

If we calculate the expression to the right of the equals sign, we get 27, which is the decimal form of the binary number 11011. For the inverse process, we successively divide the decimal number by 2 (the binary base), and we make a note of the remainders until we obtain a coefficient of 1. The binary number will have the final coefficient as its first digit, and this will be followed by the remainders starting with the last in the list. To visualise this process, we will write the number 76 in binary:

76 divided by 2 has a coefficient of 38 and a remainder of 0.
38 divided by 2 has a coefficient of 19 and a remainder of 0.
19 divided by 2 has a coefficient of 9 and a remainder of 1.
9 divided by 2 has a coefficient of 4 and a remainder of 1.
4 divided by 2 has a coefficient of 2 and a remainder of 0.
2 divided by 2 has a coefficient of 1 and a remainder of 0.

Therefore, the number 76 written in a binary system would be 1001100. This result can be verified in the previous ASCII table (keep in mind that in the corresponding code we include an additional 0 at the beginning to create strings of four digits). Converting a quantity expressed in one numeral system to another is called a base change.

Codes for detecting transmission errors

The codes outlined above make it possible for secure and effective communications between computers, between programs and between users. But this on-line language is based on a general theory of information that underlies the process of communication itself. The first step in formulating this theory is so basic that it is sometimes easy to overlook: how to measure information.

A phrase as simple as "2 kB attachment" is based a long series of brilliant intuitions that start with an article published in two parts in 1948 by the American engineer, Claude E. Shannon, and titled *A Mathematical Theory of Communication*. In this seminal article, Shannon proposed a unit of measurement for the quantity of information that he called a bit. The general problem that led to Shannon's work was one that will be familiar to modern readers: What is the best way to encrypt a message to prevent it being corrupted during transmission? Shannon concluded that it was impossible to define a code that would always prevent the loss of information. Put another way, errors will inevitably occur when information is

transmitted. However, this conclusion did not halt efforts to define standards of codification that, even if they could not prevent corruption, could at least ensure the highest levels of reliability.

In digital transmission of information, once a message has been generated by the sender (that can easily be a non-human agent, such as a computer or some other device), it is encrypted in a binary system and enters a channel of communication that consists of the sender's computer and that of the receiver plus the connection itself, which is either a physical cable or wireless (radio waves, infrared etc) The journey through the channel is the most sensitive process because the message can be subjected to all kinds of interference, including mixing with other signals, the adverse affects of temperature in the physical medium, and attenuation (weakening) of the signal as it passes through the medium These sources of interference are termed *noise*.

To minimise the impact of noise, not only do you have to protect the connection, you also have to establish a way of detecting errors and correcting them when they arise.

One of these methods is called redundancy Redundancy consists of the repetition, under determined criteria, of certain characteristics of the message Here is an example that will help to clarify the process Let us imagine text in which each word is made up of four bits, for a total of 16 words ($2^4 = 16$), each one of the type $a_1a_2a_3a_4$ Before sending a message we add three additional bits to the word $c_1c_2c_3$, so that the encoded message as it travels through the communication channel will have the form $a_1a_2a_3a_4c_1c_2c_3$ The elements $c_1c_2c_3$ will ensure the security of the message – they are called parity codes – and they are generated as follows.

$$c_1 = \begin{cases} 0 \text{ if } a_1 + a_2 + a_3 \text{ is even} \\ 1 \text{ if } a_1 + a_2 + a_3 \text{ is odd} \end{cases}$$

$$c_2 = \begin{cases} 0 \text{ if } a_1 + a_2 + a_4 \text{ is even} \\ 1 \text{ if } a_1 + a_2 + a_4 \text{ is odd} \end{cases}$$

$$c_3 = \begin{cases} 0 \text{ if } a_2 + a_3 + a_4 \text{ is even} \\ 1 \text{ if } a_2 + a_3 + a_4 \text{ is odd} \end{cases}$$

We would assign the following parity codes to the message 0111:

Since $0+1+1=2$ even, the number $c_1=0$

Since $0+1+1=2$ even, the number $c_2=0$

Since $1+1+1=3$ odd, the number $c_3=1$

Consequently, the message 0111 would be transmitted as 0111001. From the following 16 "words" we thus get the table:

Original message	Sent message
0000	0000000
0001	0001011
0010	0010111
0100	0100101
1000	1000110
1100	1100011
1010	1010001
1001	1001101
0110	0110010
0101	0101110
0011	0011100
1110	1110100
1101	1101000
1011	1011010
0111	0111001
1111	1111111

GENIUS WITHOUT A PRIZE

Claude Elwood Shannon (1916–2001) was one of the greatest scientific figures of the 20th century. Educated in electrical engineering at the University of Michigan and the Massachusetts Institute of Technology, he worked as a mathematician at Bell Labs where he did research on cryptography and communication theory. His contributions to information theory are sufficient to place him at the top table of innovators, but since his work was halfway between mathematics and information technology, he never received the prize coveted by all scientists: the Nobel

Let us suppose that at the end of the journey, the receiving system gets the message 1010110. Note that this combination of 0 and 1 is not among the possible messages and must, therefore, be a transmission error. To try to correct the error, the system compares each digit with the set of digits of possible messages to find a more probable alternative. To do so, it checks how many of the digits appear to be wrong, as we show below:

Possible message	0000000	0001011	0010111	0100101	1000110
Received message	1010110	1010110	1010110	1010110	1010110
Number of different digits in each position	4	5	2	5	1

Possible message	1100011	101[illegible]01	1001101	0110010	0101110
Received message	1010110	1010110	1010110	1010110	1010110
Number of different digits in each position	4	3	4	3	4

Possible message	[illegible]	1110100	1101000	1011010	0111001	1111111
Received message	1010110	1010110	1010110	1010110	1010110	1010110
Number of different digits in each position	3	2	5	2	6	3

The erroneous word (1010110) differs from another word (1000110) by a single digit. Since the difference is the smallest, the system will offer the recipient this second, corrected version. The principle is analogous to that of the spell checker on a word processor. When it detects a term that does not register in its internal dictionary, it proposes a series of close alternatives. The number of positions by which a message, understood as a sequence of characters, differs from another is known as the *distance between two sequences*. This specific mechanism of error detection and error correction was proposed by the American Richard W. Hamming (1915–1998), a contemporary of Claude Shannon.

In information, as in any other field, it is one thing to detect the possible errors, and quite another to correct them. In encryptions, such as this last example, if there is only one candidate of minimal distance, the problem is simple enough. If we count the minimum number of times that 1 appears in the sequence (omitting the sequence that is all 0), we can verify that:

If t is odd, we can correct $\dfrac{t-1}{2}$ errors.

If t is even, we can correct $\dfrac{t-2}{2}$ errors.

If our only purpose is to detect errors, the maximum number we can detect will be $t-1$. In the 16-character language expounded before, $t = 3$, from which we get that the mechanism is capable of detecting $3-1=2$ errors, and to correct $(3-1):2=1$ error.

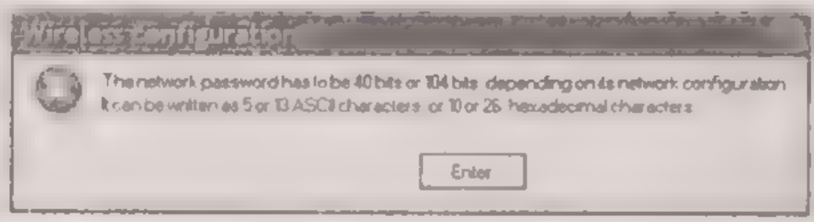
THIRD GENERATION CRYPTOGRAPHY

In 1997, a protocol was introduced for the secure transmission of information through wireless networks by the name of WEP, the acronym for *Wired Equivalent Privacy*. This protocol includes an encrypting algorithm called RC4, with two types of codes of 5 and 13 ASCII characters respectively. We are dealing, therefore, with codes of 40 or 104 bits or, alternatively, of 10 or 26 hexadecimal characters

5 alphanumeric letters = 40 bits = 10 hexadecimal characters

13 alphanumeric letters = 104 bits = 26 hexadecimal characters

The connection provider supplies the codes, although the user can generally change them. Before establishing the connection, the computer asks for the key. In the following dialogue box we see an error message asking for the WEP key, specifying its length in bits, ASCII characters and hexadecimal characters



In truth, the real keys are longer. Starting with those supplied by the user, the algorithm RC4 generates a new key with more bits, which is the one used to cipher the transmission. This is public key cryptography and it will be explained in more detail in Chapter 5. A user who wishes to change the key would do well to remember that a key of ten hexadecimal characters will be more secure than a key of five alphanumeric characters, although the bit size is the same. Of course, it is also certain that "james" is easier to remember than its hexadecimal equivalent "6A616D6573"

Other codes: the standards of industry and commerce

Although less glamorous than cryptography or binary mathematics, and often invisible to us despite their ubiquity, the standardised codes of banks, supermarkets, and other large economic players are one of the pillars that support modern society. In the case of these codes, the priority is to ensure the unique and accurate identification of products, be they bank accounts, books or apples. We will now examine them in more detail.

Credit cards

The debit and credit cards offered by major banks and department stores are essentially identified by set groups of numbers and calculated with the same algorithm and verification system, all based on our old friend, modular arithmetic. The majority of cards have 16 digits, made up of numbers between 0 and 9. The numbers are grouped in 4 digits so they can be read more easily. For our purposes we will denote them as:

ABCD EFGH IJKL MNOP

Each group of digits codifies some piece of information: the first group (ABCD) corresponds to the ID of the bank (or whichever entity is providing the service). Each bank has a different number that may vary according to the continent, and that is also related to the card's brand and conditions. For example, in the case of VISA and some prominent banks, the first four numbers are as follows.

A B C D	Provider
4941	Citibank
4024	Bank of America
4128	Citibank (USA)
4302	HSBC

The fifth digit (E) corresponds to the type of card and indicates which financial institution is administering the account:

Type	Provider
3	American Express
4 0 2	Visa
5 [illegible]	MasterCard
6	Discover

As we can see, it is not a rigid rule.

The following ten digits (FGH IJKL MNO) are a unique identifier for each card. This identification not only supplies a reference number for each client account, but it is also linked to the branding of the card – Classic, Gold, Platinum etc – and the associated credit limit, interest rates on type of balance and its expiration date.

Finally, there is a control digit (P) that relates to the previous digits according to Luhn's algorithm, so called in honour of Hans Peter Luhn, the German engineer that developed it. For a 16-digit card, this algorithm works as follows:

1) For each digit in an odd position, starting with the first number on the left, we calculate a new digit by multiplying it by two. If the result of this multiplication is greater than 9, we add the two digits of the new number (or we perform the equivalent operation of subtracting 9). For example, if we get 18, we add 1 + 8 = 9, or else we subtract 18 – 9 = 9.

2) Next, we add all the numbers calculated in this way, and the digits located in even positions (including the final control digit).

3) If the total is a multiple of 10 (that is, its value is 0 in mod 10), the numbers on the card are valid. Note that it is the final control digit that makes the eventual total a multiple of 10.

DINER'S CLUB

One of the first credit cards to gain wide acceptance was Diner's Club. The driving force behind it was the American Frank McNamara. In 1950, he managed to persuade various restaurants to accept payment by credit when offered with a personalised, guaranteed card that McNamara distributed to his best clients. The most common use of credit cards in their first decades was for American travelling salesmen to pay for meals while on the road.

For example, in the case of a card numbered as follows:

1234 5678 9012 3452

According to Luhn's algorithm:

$$
\begin{aligned}
&1\cdot 2 = 2\\
&3\cdot 2 = 6\\
&5\cdot 2 = 10 \Rightarrow 1+0 = 1\\
&7.2=14 \Rightarrow 1+4 = 5 \text{ (or } 14-9 = 5)\\
&9.2 = 18 \Rightarrow 1+8 = 9\\
&1.2 = 2\\
&3.2 = 6\\
&5.2 = 10 \Rightarrow 1+0 = 1
\end{aligned}
$$

$$2 + 6 + 1 + 5 + 9 + 2 + 6 + 1 = 32$$
$$2 + 4 + 6 + 8 + 0 + 2 + 4 + 2 = 28$$
$$32 + 28 = 60$$

The result is 60, a multiple of 10. Therefore the card's code number is valid.

Another way to apply Luhn's algorithm is as follows: the number of card ABCD EFGH IJKL MNOP is correct if the double of the sum of the digits in an odd position and the sum of the digits in an equal position plus the number of digits in an odd position that are greater than 4 is a multiple of 10. That mouthful is perhaps better expressed as $2(A + C + E + G + I + K + M + O) + (B + D + F + H + J + L + N + P) +$ the number of digits in an odd position greater than $4 \equiv 0 \pmod{10}$.

Applying this second version of the algorithm to the earlier example:

1234 5678 9012 3452

$$2\cdot(1 + 3 + 5 + 7 + 9 + 1 + 3 + 5) + (2 + 4 + 6 + 8 + 0 + 2 + 4 + 2) + 4 =$$
$$= 100 \equiv 0 \pmod{10}.$$

Again we have verified that the number is a valid credit card number and have shown that apparently random card codes follow a strict mathematical standard.

EXCEL APPLICATION FOR THE CALCULATION OF THE CONTROL DIGIT OF A CREDIT CARD

The number associated with a credit card [illegible] code. The numbers are grouped in four sets of four digits. The [illegible] according to the algorithm below

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
2																					C.D..
3	**Credit card no.**	5	5	2	1		4	5	7	2		6	1	6	2		3	6	2	4	
4																					
5	**Digits used**		5	2	1		4	5	7	2		6	1	6	2		3	6	2		
6	Digits in even position			2			4		7			6		6			3		2		
7	Sum of digits in even position																			30	
8	Number of digits in even position greater than 4																			3	
9	Sum of the two previous quantities																				33
10	Digits in odd position		5		1			5		2			1		2			6			
11	Sum of digits in odd position																				22
12	Sum of the two preceding results plus 1																				56
13	Remainder of dividing the previous result by 10																				6
14	The C.D. is 0 if the previous result is 0, otherwise it is 10 less the previous result																				4

Would it be possible to recover a digit missing from a card code? Yes, as long as we are dealing with a valid credit card. Let us solve the value of X in the number 4539 4512 03X8 7356.

We start by multiplying by 2 the numbers in the odd positions (4-3-4-1-0-X--7-5), reducing them to a single digit.

$$4 \cdot 2 = 8$$
$$3 \cdot 2 = 6$$
$$4 \cdot 2 = 8$$
$$1 \cdot 2 = 2$$
$$0 \cdot 2 = 0$$
$$X2 = 2X$$
$$7 \cdot 2 = 14,\ 14 - 9 = 5$$
$$5 \cdot 2 = 10,\ 10 - 9 = 1.$$

We add the digits of the even positions and the new digits from the odd positions and we get:

$$30+41+2X=71+2X$$

71+2X, which we know has to be a multiple of 10.

If the value of X were greater than 4 and less than 10, 2X would be a number between 10 and 18. The value of 2X reduced to a single digit is

2X − 9, so the previous sum would be 71 + 2X − 9. The only value of X that would make the expression a multiple of 10 is 9. If, on the contrary, X were less than or equal to 4, we see that there is no value that proves that 71 + 2X is a multiple of 10.

Consequently, the lost digit is 9, and the complete number of the credit card is 4539 4512 0398 7356.

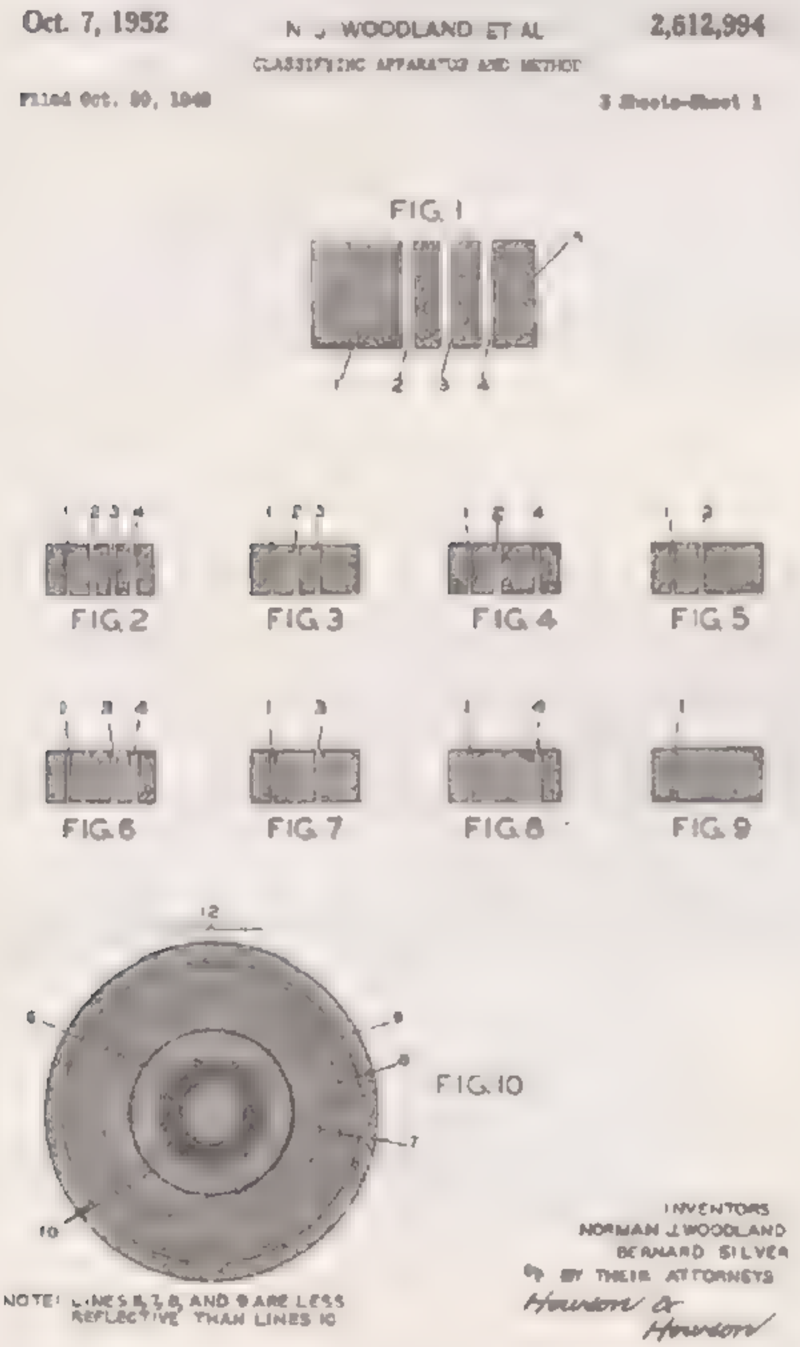
Barcodes

The first barcode system was patented on October 7, 1952, by Americans Norman Woodland and Bernard Silver. The early codes were quite different from todays. In place of the familiar bars, Woodland and Silver thought in terms of concentric circles. The first official use of a barcode in a shop was in 1974 in Troy, Ohio.

The modern barcode consists of a series of black bars (which are coded as 1 in the binary system) and the blank spaces between them (which are coded as 0). Barcodes are used to identify physical objects. The codes are generally printed on labels and are read by an optical device. This device, similar to a scanner, measures the reflected light and converts bands of areas of dark and light into an alphanumeric key, which it then sends to a computer. There are numerous standards for barcodes

How the thickness of bars and spaces in a barcode correspond to binary digits.

Code 128, Code 39, Codabar, EAN (this appeared in 1976 in versions of 8 and 13 digits) and UPC (Universal Product Code, used primarily in the US and available in versions of 12 and 8 digits). The most common code is the 13 digit version of EAN. Despite the variety in standards, the barcode allows for any product to be identified in any part of the world, swiftly and without a large margin of error.



The patent of Woodland and Silver's system of concentric circles that pre-dates modern barcodes

EXCEL APPLICATION FOR THE CALCULATION OF THE CONTROL DIGIT OF THE EAN-13 CODE

A barcode [illegible] EAN [illegible] 13 digits [illegible] called a control digit (C.D.).

The 13 digits are distributed in four groups

Country		Company					Product					C.D
8	4	1	1	3	4	9	0	4	5	1	2	6

The [illegible]

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
[illegible]	Country			Company						Product						C.D
4	3	4		1	[illegible]	[illegible]	5	[illegible]		1	2	[illegible]	[illegible]	2		[illegible]
5																
[illegible]	Sum of digits in odd position															27
[illegible]	Sum of digits in even position and the result multiplied by 3															51
8	Sum of the two previous results															78
9	Remainder of dividing the previous result by 10															8
10	The C.D. is 0 or 10 less the previous result															2

[illegible]

	Country			Company						Product						C.D
	3	4		1	[illegible]	3	5	9		1	2	5	9	2		=R10
	Sum of digits in odd position															=C4+F4+H4+J4+M4+O4
	Sum of digits in even position and the result multiplied by 3															=(D4+G4+I4+L4+N4+P4)*3
	Sum of the two previous results															=R6+R7
	Remainder of dividing the previous result by 10															=RESIDUO(R8,10)
	The C.D. is 0 or 10 less the previous result															[illegible] R9

The EAN-13 barcode

The EAN code, originally named as the acronym for "European Article Number" when it was created in 1976, is now known as the International Article Number. It is the most established barcode standard and is used throughout the world. EAN codes generally consist of 13 digits represented by black bars and white spaces that together form a binary code that is easy to read. EAN-13 represents these 13 digits by means of 30 bars and spaces. The digits are distributed in three parts: the first one, that consists of 2 or 3 numbers, indicates the country code; the second, made up of 9 or 10 numbers, identifies the company and the product; the third, of only one digit, acts as the control code. For a code ABCDEFGHIJKLM these parts are divided as follows:

- •The first two (AB) form the code of the country of origin of the product. The UK's code is 50, for example, while Ireland's is 539.
- •The next five (CDEFG) identify the company producing the product.
- •The other five (HIJKL) indicate the product code that has been assigned by the company.
- •The last (M) is the control number. To calculate it, we have to add the digits in the odd positions, starting at the left and without counting the control number. To the resulting value we then add three times the sum of the digits in even positions. The control number is the value that makes the total sum just calculated a multiple of 10. As we can see, the barcode control system is strongly reminiscent of the one employed by credit cards.

8 413871 003049

Let's verify if this barcode is valid:

8413871003049

$$8+1+8+1+0+0+3(4+3+7+0+3+4)=18+3(21)=18+63=81.$$

The correct control digit should be $90-81=9$.

The mathematical model of the algorithm is based on modular arithmetic (modulus 10) as follows:

ABCDEFGHIJKLM, we will call the value of the expression N

$$A+C+E+G+I+K+3(B+D+F+H+J+L)=N$$

and n the value of N in modulus 10. The control digit M is defined as $M=10-n$. In our example, we have that $81 \equiv 1 \pmod{10}$, therefore the control digit will, indeed, be $10-1=9$.

The previous algorithm can be formulated in an equivalent way using the control digit in the calculations. The following technique allows us to verify the validity of the control code without having to calculate it first.

$$A+C+E+G+I+K+3(B+D+F+H+J+L)+M \equiv 0 \pmod{10}$$

For the sample code

5701263900544

$$5+0+2+3+0+5+3(7+1+6+9+0+4)+4=100.$$

$$100 \equiv 0 \pmod{10}.$$

The code is therefore valid.

Out of curiosity, we will try to determine the value of a lost number of a barcode. Specifically, that represented by X in the following code

401332003X497

We arrange the numbers according to the algorithm

$$4+1+3+0+3+4(0+3+2+0+X+9)+7=64+3X\equiv 0 \pmod{10}.$$

In modulus 10, we get the following equation:

$$4+3X\equiv 0 \pmod{10}.$$

$$3X\equiv -4+0\equiv -4+10\cdot 1=6 \pmod{10}.$$

Note that 3 has an inverse since $\gcd(3,10)=1$.

We therefore find that X has to be 2. Therefore the valid code is

$$4013320032497.$$

QR CODES

In 1994, the Japanese company Denso-Wave developed a graphic system of encryption to identify the parts of cars in an assembly line. The system, called QR for the speed with which it could be read by machines designed for the purpose (the initials QR stand for *quick response*), ended up expanding way beyond car factories. In just a few years, the majority of Japan's mobile telephones could instantly read the information contained in the code. The QR is a type of matrix code, formed by a variable number of black or white squares that, in turn, are arranged in the shape of a larger square. The squares represent a binary value (0 or 1); therefore, they operate in a similar way to bar codes, although adding a second dimension gives the code a larger storage capacity.

A QR Code of 37 rows for the University of Osaka, Japan

Chapter 5

An Open Secret: the Cryptography of Public Keys

Cryptography was not ignored during the rapid growth in computing technology. To use a computer to cipher a message is more or less the same process as ciphering without a computer, but there are three fundamental differences. First, a computer can be programmed to simulate the work of a conventional encoding machine of, for example, 1,000 rotors without the need to physically build such a device. Second, a computer works only with binary numbers and, therefore, all ciphering will occur at this level—even if the numerical information is subsequently deciphered into text again. And third, computers are extremely fast at computing ciphers and deciphering messages.

The first ciphers designed to take advantage of the potential of computers were developed in the 1970s. An example is Lucifer, a cipher that divided the text into blocks of 64 bits and encrypted some of them by means of a complex substitution and then grouped them again into a newly ciphered block of bits and continued to repeat the process. The system required both sender and receiver to be equipped with a computer running the same encryption program and a shared numerical key. A 56-bit version of Lucifer called DES was introduced in 1976. DES, standing for Data Encryption Standard, is still in use today although it was cracked in 1999 and largely replaced by the 128-bit AES (Advanced Encryption Standard) in 2002.

Without a doubt, this encryption made the most of a computer's processing power, but just like their thousand-year-old predecessors, computerized codes were still vulnerable to the danger that an unauthorised receiver could obtain the codes and, knowing the encryption algorithm, decipher the message. This basic weakness of every "classical" system of cryptography is known as *the key distribution problem*.

The key distribution problem

It is generally agreed that encryption keys should be protected more than the algorithm in order to maintain the security of a code. That creates a problem: how to

distribute keys securely. Even in the simplest cases, it could lead to difficult logistical problems, such as how to distribute thousands of code books among a large army, or how to distribute them to mobile communication centres that operate in extreme circumstances, like submarine crews or units in the heat of battle. No matter how sophisticated a classical encryption system was, all were vulnerable to the interception of their respective keys.

The Diffie-Hellman algorithm

The notion of a secure exchange of keys might sound self-contradictory: how can you send a key as a message which has already been encrypted – with the key exchanged previously in the usual way? However, if the exchange is set up as a communication with multiple exchanges, one can imagine a solution to the problem – at least on a theoretical level.

Let us suppose that a sender named James encrypts a message with his key and sends it to the receiver Peter. The latter re-encrypts the ciphered message with his key and returns it to the sender. James deciphers the message with his key and sends this new message, that is now only ciphered with Peter's key, who goes on to decipher it. The age-old problem of the secure exchange of keys has all of a sudden been resolved! Can this really be true? Sadly, no. In any complex encryption algorithm, the order in which the keys are applied is critical, and we have seen that in our theoretical example, James has to decipher a message that has already been ciphered with another key. When the order of the ciphers is reversed, the result will

THE MEN BEHIND THE ALGORITHM

Bailey Whitfield Diffie (left) was born in 1944 in the United States. With a mathematics degree from the Massachusetts Institute of Technology (MIT), he served as the Chief Security Officer and Vice President of California-based Sun Microsystems from 2002 until 2009. For his part, the engineer Martin Hellmann was born in 1945 and carried out his professional career at IBM and MIT, where he collaborated with Diffie.

be gibberish. The theory is not really explained by the above scenario, but it shines a light on a way to solve the problem. In 1976 two young American scientists, Bailey Whitfield Diffie and Martin Hellman, found a way in which two people could exchange ciphered messages without having to exchange any secret key whatsoever. This method makes use of modular arithmetic, as well as the properties of prime numbers. The ideas is as follows:

1. James picks a number that he keeps secret. We will call this number N_{j1}.
2. Peter picks another random number that he, too, keeps secret. We will call this number N_{p1}.
3. Next, both James and Peter apply a function of the type $f(x) = a^x \pmod p$ to their respective numbers, with p being a prime number known by both.
   - From this operation James obtains a new number, N_{j2}, which he then sends to Peter.
   - Performing the same operation, Peter obtains a new number, N_{p2}, which he sends to James.
4. James solves an equation of the form $N_{p2}^{N_{j1}} \pmod p$ and gets a new number, C_j.
5. Peter solves an equation of the form $N_{j2}^{N_{p1}} \pmod p$ and gets a new number, C_p.

Although it appears impossible, C_j and C_p are the same. And now we have the key. Note that the only times in which James and Peter exchanged information was when they agreed on the function $f(x) = a^x \pmod p$ and when they sent each other N_{j2} and N_{p2}. Neither are the key and their interception, therefore, will not threaten the security of the encryption system. The key of this system will have the general form:

$$a^{N_{j1} N_{p1}} \text{ in modulus } p.$$

It is also important to take into account that the original function has the special feature of not being reversible, that is, knowing both the function and the result of applying it to a variable x, it is impossible (or at least very difficult) to obtain the original variable x.

Next, and to emphasise the point, we will repeat the process with specific values. The function being used is:

$$f(x) = 7^x \text{ (mod. 11).}$$

1. James picks a number N_J, for example 3, and calculates $f(x) = 7^x$ (mod 11), obtaining $f(3) = 7^3 \equiv 2$ (mod. 11).
2. Peter picks a number N_P, for example 6, and calculates $f(x) = 7^x$ (mod 11) obtaining $f(6) = 7^6 \equiv 4$ (mod. 11).
3. James sends Peter his result, 2, and Peter does the same with his, 4.
4. James calculates $4^3 \equiv 9$ (mod. 11).
5. Peter calculates $2^6 \equiv 9$ (mod. 11).

This value, 9, will be the key of the system.

James and Peter have exchanged both the function $f(x)$ and the numbers 2 and 4. Is this information useful to an eavesdropper? Let us suppose that our unwanted recipient knows both the function and the numbers. His problem now is to solve N_J and N_P in modulus 11 N_J and N_P being the numbers that both James and Peter keep secret – even from each other. If the spy manages to discover them, he would have the key only to solve a[illegible] in modulus p. The solution to this problem by the way, is termed a discrete logarithm in mathematics. For example, in the case of:

$$f(x) = 3^x \text{ (mod. 17)}$$

we can see that $3^x = 15$ (mod 17) and trying different values of x, we find that x = 6 and verify the relation $3^x = 15$.

The algorithms of this type, and the problem of the discrete logarithm did not receive much attention until the beginning of the 1990s and it has only been in recent years that it has been developed. In the example above, we say that 6 is the discrete logarithm of 15 with a base of 3 with modulus 17

The special characteristic of this type of equations is, as we have already mentioned, that they are difficult to reverse – they are *asymmetrical* For values of p greater than 3[illegible] and of a greater than 100, the solution – and, therefore, the cracking of the key – becomes extremely difficult.

> ### VIRUSES AND "BACK DOORS"
>
> Even the most secure [illegible] key [illegible] sequently [illegible] computer [illegible] It wreaks hav[illegible] eager [illegible] deleted the [illegible] key [illegible] and [illegible] communications between the infected computers

This algorithm is the foundation of modern cryptography. Diffie and Hellman presented their idea at the National Computer Conference, in a seminar that can only be termed as groundbreaking. Their paper can be examined in its entirety at www.cs.berkeley.edu/~christos/classics/diffiehellman.pdf, where it appears with the title *New Directions in Cryptography*.

Diffie-Hellman's algorithm demonstrated the possibility of creating a cryptographic method that did not require the exchanging of keys yet, paradoxically, relied on public communication for part of the process – the initial pair of numbers that serve to determine the key.

Put another way, it made it possible to have a secure encryption system between senders and receivers who never had to meet to agree on a key in secret. However, certain problems remained. If James wants to send Peter a message while Peter is asleep, for example, he will have to wait for his opposite number to wake up to carry out the process of generating the key.

In the process of trying to discover new, more effective algorithms, Diffie theorised a system in which the ciphering key would be different from the deciphering key, and therefore one could never derive one from the other. In this theoretical system, the sender would have two keys: the encrypting key and the decrypting key. Of the two, the sender would only make the first one *public* so that whoever should want to send him a message could encrypt it. Having received the message, the sender would go on to decipher it using the decrypting key that had obviously remained secret. Would it be possible to put such as system into practice?

The primes come to the rescue: the RSA algorithm

In August of 1977, the famous US science writer, Martin Gardner, entitled his column on recreational mathematics for the journal *Scientific American*, "A new kind of cipher that would take millions of years to break." After explaining the principles of the public key system, he listed the ciphered message as well as the public key N used to create the cipher:

N = 114.381.625.757.888.867.669.235.779.976.146.
612.010 218 296 721 242 362 562 561 842 935 706 935 245 733
897 830 597 123 563 958 705 058 989 075 147 599 290 026 879
543.541.

Gardner challenged his readers to decipher the message from the information given, and even offered up a clue – the solution required that N be factorized into its prime components p and q. To top it off, Gardner promised a prize of $100 (a reasonable sum at the time) to whoever got the right answer first. Anyone wanting more information on the cipher, Gardner wrote, should send a request to the cipher's creators, Ron Rivest, Adi Shamir and Len Adelman from MIT's Laboratory for Information.

The correct answer was not received until 17 years later, and to find it took the collaboration of more than 600 people. The keys turned out to be p = 32 769 132 993 266 709 549.961 988 190 834 461 413 177 642 967 992 942 539 798 288 533 and q = 3 490 529 510 847 650 949 147 849 619 903 898 133 417 764 638 493,387 843 990 820 577, and the ciphered message, "The magic words are squeamish ossifrage."

The algorithm Gardner presented is known as RSA, an acronym from the surnames Rivest, Shamir and Adelman. It is the first practical implementation of the public key model posited by Diffie, and it is regularly used today. The security it offers is almost total because the deciphering process is incredibly hard work, although not impossible. Next, we will look at the basis of the system in simplified form.

The RSA algorithm in detail

The RSA algorithm is based on certain properties of prime numbers that the interested reader can find in the Appendix. We will limit ourselves here to setting out the basic assumptions that underlie it.

- The group of numbers smaller than n that are also prime with n are called Euler's function and are expressed as $\varphi(n)$.
- If $n = pq$ given that p and q are prime numbers, then $\varphi(n) = (p-1)(q-1)$.
- From "Fermat's Little Theorem" we know that if a is a whole number larger than 0 and p a prime number, we have to have $a^{p-1} \equiv 1 \pmod{p}$.
- According to Euler's theorem, if $gcd(n,a) = 1$, then $a^{\varphi(n)} \equiv 1 \pmod{n}$.

As mentioned, the system is described as "public key" because the encryption key is given to any sender interested in transmitting messages. Each recipient has his own public key. The messages will always be transmitted translated into numbers, be it as ASCII code or any other system.

First, James generates a value n as a product of two prime numbers p and q ($n = p \cdot q$) and we pick a value e so that the $gcd(\varphi(n), e) = 1$. Remember that $\varphi(n) = (p-1)(q-1)$. The data that is made public is the value of n and the value of e; under no circumstances will we provide the values p and q. The pair (n,e) is the public key of the system, and the values p and q are known as RSA numbers. In parallel, James calculates the only value of d in modulus $\varphi(n)$ that satisfies that $d \cdot e \equiv 1$, that is, the inverse of e in modulus $\varphi(n)$. We know that this inverse exists because $gcd(\varphi(n), e) = 1$. This value d is the private key of the system. For his part, Peter uses the public key (n,e) to encrypt message M by means of the function $M = m^e \pmod{n}$. Having received the message, James carries out the operation $M' = (m^e)^d \pmod{n}$. This expression is equivalent to $M' = m^{ed} = m \pmod{n}$, which proves that the message can be deciphered.

We will now apply this procedure with specific numerical values.

If $p = 3$ and $q = 11$ we have $n = 33$, $\varphi(33) = (3-1)(11-1) = 20$. James picks e that does not have a divisor in common with 20, for example $e = 7$. James's public key is (33,7).

• Meanwhile, James has calculated a private key d that will be the inverse of 7 mod 20, that is $7 \cdot 3 \equiv 1$ (mod. 20), and therefore $d = 3$.

• Peter acquires the public key and wishes to send us the message "9". To cipher it, he uses James's public key and solves:

$$9^7 = 4.782.969 \equiv 15 \text{ (mod. 33)}.$$

The ciphered message is 15. Peter sends us the message.

James receives the message 15, and deciphers it:

$$15^3 = 3.375 \equiv 9 \text{ (mod. 33)}.$$

The message has been correctly deciphered.

As we pick larger prime numbers p, q, the difficulty of implementing the RSA algorithm increases to a point where the use of a computer for the calculation of the solutions becomes necessary. For example, if $p = 23$ and $q = 17$, then $n = 391$. The public key that results for $e = 3$ is (391,3). Consequently, $d = 235$. For a plaintext message like 34, the deciphering operation is:

$$204^{235} \equiv 34 \text{ (mod. 391)}.$$

Take note of the order of magnitude and imagine the gigantic calculation capacity necessary to find this solution.

Why should we trust in the RSA algorithm?

A potential spy knows the values of n and of e because they are public. To decipher the message he will need, along with the value of d, the private key. As we demonstrated in the preceding example, the value d is generated from n and from e. Where does the security stem from? Let us remember that to construct d, it is necessary to know $\varphi(n) = (p-1)(q-1)$, in particular, p and q. For this, it is "sufficient" to decompose n as a product of two prime numbers p and q. The problem for the spy is that to factorize a large number as a product of two prime numbers is a slow and laborious process. If n is sufficiently large (of the order of more than [illegible] digits), there is no known way to find p and q in a reasonable amount of time.

Today, the prime numbers used in the ciphering of messages of the most sensitive nature exceed 200 digits.

Reasonable privacy

The RSA algorithm consumes a great deal of computing time and requires high-powered processors. Until the 1980s, only governments, the military, and large enterprises had sufficiently powerful computers to work with RSA. As a result, they enjoyed a *de facto* monopoly over effective encryption. In the summer of 1991, Philip Zimmermann, an American physicist and activist for privacy, offered free of charge the PGP (Pretty Good Privacy) system, an encryption algorithm capable of working on home computers. PGP employs the classic symmetrical codification – which gives it greater speed on home computers – but it ciphers the keys with an asymmetrical RSA.

Zimmermann explained the reasons for this measure in an open letter that deserves to be quoted here, at least partially, for its prescient description of the way we live, work and communicate two decades later.

> "It's personal. It's private. And it's no one's business but yours. You may be planning a political campaign, discussing your taxes, or having a secret romance. Or you may be doing something that you feel shouldn't be illegal, but is. Whatever it is, you don't want your private electronic mail or confidential documents read by anyone else. There's nothing wrong with asserting your privacy. Privacy is as apple-pie as the Constitution...

> "We are moving toward a future when the nation will be crisscrossed with high capacity fibre-optic data networks linking together all our increasingly ubiquitous personal computers. E-mail will be the norm for everyone, not the novelty it is today. The government will protect our E-mails with Government-designed encryption protocols. Probably most people will acquiesce to that. But perhaps some people will prefer their own protective measures... If privacy is outlawed, only outlaws will have privacy.

> Intelligence agencies have access to good cryptographic technology. So do the big arms and drug traffickers. So do defense contractors, oil companies, and other corporate giants. But ordinary people and grassroots political or-

SECURITY FOR EVERYONE

Philip Zimmermann, born in 1954, is an American physicist and software engineer who has spearheaded a movement that aims to make modern cryptography available to all. Besides launching the PGP system, in 2006 he created Zfone, a software program for secure voice communication over the internet, and he is president of the Open PGP Alliance, a lobby group in favour of open code software.

ganizations mostly have not had access to affordable military-grade public-key cryptographic technology. Until now.

PGP empowers people to take their privacy into their own hands. There's a growing social need for it. That's why I wrote it."

From Zimmermann's reflections, we can see that the price of living during the information age is to have our traditional notions of privacy threatened. Consequently, a good understanding of the codification and encryption mechanisms that surround us would not only make us wiser, but could also prove to be enormously useful when it comes to protecting what is valuable to us.

The use of PGP has been spreading since its creation and constitutes the most important private cryptography tool available today.

Authentication of messages and keys

The different systems of public key encryption – or public and private keys combined, like PGP – ensure a high level of confidentiality in the transmission of information. However, the security of a complex communication system like the Internet does not reside solely in confidentiality.

Before the arrival of modern communication technologies, the vast majority of messages originated from known sources, such as family, friends, or a handful of professional relationships. Today, however, each individual is bombarded by an avalanche of communications from a myriad of sources. The authenticity of these

communications can frequently be impossible to determine just by reading them, with all the problems that derive from that. For example, how can we prevent someone falsifying the address of origin of an email?

Diffie and Hellman themselves proposed an ingenious way to use public key encryption to authenticate the origin of a message. In a cryptography system of this type, the sender ciphers the message with the public key of the receiver, who in turn uses his own private key to decipher the message. Diffie and Hellman noticed that RSA and other similar algorithms displayed an interesting symmetry. The private key could also be used to cipher a message, and the public to decipher it. This operation provides no security whatsoever – the public key is easily available to everyone – but it does assure the receiver that the message comes from a specific sender, the owner of the private key. To authenticate the sender of a message it is sufficient, in theory, to add an additional encryption to the normal one with the following process:

1 The sender encrypts a message with the receiver's public key. This first step ensures confidentiality.

2 The sender again encrypts the message, this time with his private key. In this way the message is authenticated or "signed".

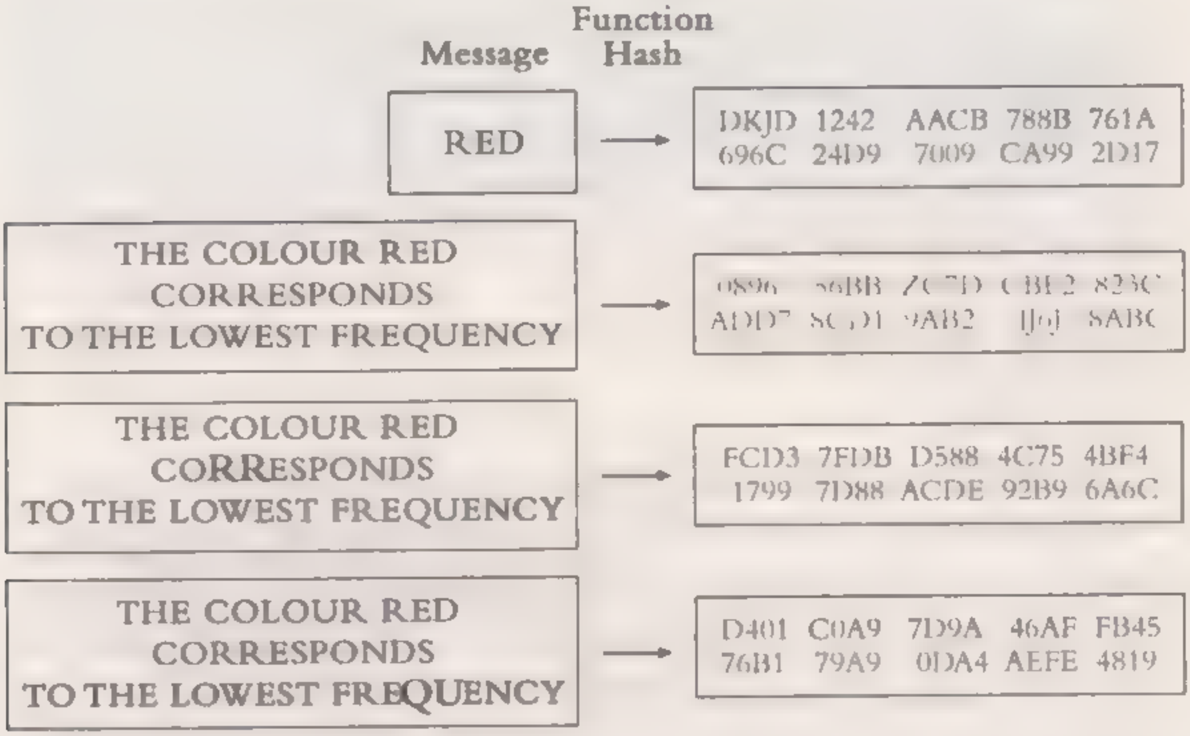
3 The recipient uses the sender's public key to undo the encryption of step 2. Thus the origin of the message is verified.

4 The receiver now uses his private key to undo the encryption of step 1.

Hash functions

One of the problems with the theoretical outline above is that the encryption of the public key requires a considerable computational capacity and to repeat the process for the purpose of signing and verifying every message would be extremely time consuming. That is why, in practice, the signing of a message is carried out by mathematical resources known as *hash* functions. Starting with the original message, these algorithms generate a simple chain of bits (usually 160), called hash, and they do it in such a way that the probability that different messages are associated with the same hash is almost zero. Also, it is practically impossible to undo the process and obtain the original message when only starting with its hash. The hash of any message is encrypted by the sender with his private key and it is sent along with the ciphered message in the conventional manner. The receiver decrypts the

message that contains the hash with the sender's public key Next, and given that he knows the hash function used by the sender, he applies that function to the message and compares the two hashes If they match, the identity of the sender is verified and, moreover, it is certain that no one else has handled the original message.



Tiny changes in the content of the message generate totally different "hashes" In this way the receiver can be sure that the text has not been manipulated

Certificates of public keys

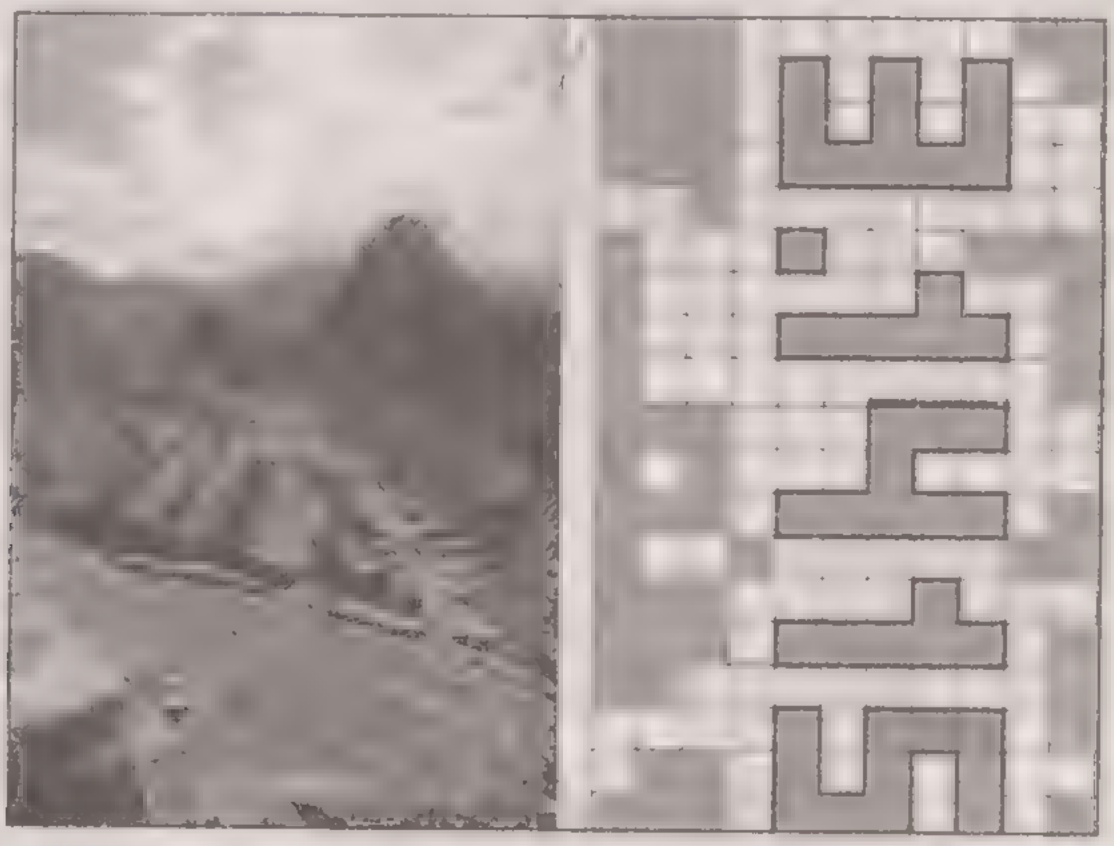
However, the most important problem to be confronted in a public key cryptography system is found, not in the authentication of the messages, but rather in that of the public keys themselves How do the sender and the recipient know that the public key of the other is valid? Let us suppose that a spy deceives the sender by giving him his own public key while making him believe that it is the receiver's key If the spy manages to intercept a message he can now use his private key to decrypt it To avoid being discovered, the spy uses the public key of the receiver to re-encrypt the message and send it to its original destination

This is why there are both public and private institutions devoted to the inde-

pendent certification of public keys. A certificate of this type contains, besides the corresponding key, information on the receiver and an expiration date. The holders of these types of keys make their certificates public, and they can now be used and exchanged with a certain degree of security.

DIGITAL STEGANOGRAPHY

[illegible] 44.1 kHz [illegible] used to transmit hidden information



An example of digital steganography [illegible] the pixels extracted from one small area that conceals the number 3 1415

But is it safe to buy on the Internet?

Most on line spies and hackers have little interest in the messages exchanged by ordinary people, with one notable exception the numbers of their credit cards The cryptography system that protects the transmission of such a sensitive piece of information (or "layer" in information science jargon is known as TLS (Transport Layer Security) It was developed by the Internet software corporation Netscape in 1994 and was adopted as the global standard two years later

The TLS protocol combines public and symmetrical keys in a rather complex process that is presented here in summary form First, the web browser of the online purchaser verifies that the online seller has a valid public key certificate If so, he uses this public key to encrypt a second key, this one symmetrical, that he sends to the seller The seller then uses his private key to decrypt the message and get the symmetrical key, which will be the one used to cipher the all the information being processed As a consequence to acquire the credit card number in any online transaction, the spy will have to penetrate not one, but two encryption systems.

Chapter 6
A Quantum Future

According to Philip Zimmermann (see Security for Everyone, page [illegible]), in Simon Singh's book *The Code Book*, "In modern cryptography it is possible to create ciphers that really are beyond the reach of all the known forms of cryptanalysis." As we have noted, to break encryption algorithms like RSA or DES and even mixed systems like PGP by brute force is beyond the computing capacity of the fastest of computers. Is it conceivable that some type of mathematical short cut could allow future spies to reduce the complexity of cryptanalysis? Although this possibility cannot be discarded, no one considers it very probable.

Is Zimmermann right? Has the thousand-year-old conflict between cryptographers and cryptanalysts been resolved?

Quantum computing

The answer is not exactly. In the last decades of the 20th century, quantum computing, a new and revolutionary way of designing and operating computers, appeared. Although still only at the theoretical stage, a quantum computer could have the calculating power to decipher today's encryption algorithms by trial and error. Cryptanalysis may be back in the game one day.

This embryonic technological revolution is based on *quantum mechanics*, a theoretical edifice erected at the start of the last century by scientists including the Dane, Niels Bohr (1885–1962), the Briton Paul Dirac (1902–1984), and the Germans Max Planck (1858–1947), Werner Heisenberg (1901–1976) and Erwin Schrödinger (1887–1961), among many others. The vision of the universe postulated by quantum mechanics is so profoundly counter-intuitive that Albert Einstein was famously quoted in opposition to it, "God does not play dice." Despite Einstein's reservations, the theory of quantum mechanics has been successfully tested on countless occasions, and its validity is now beyond question. The scientific community as a whole assumes that at the macroscopic level – that is, the universe of the stars, of houses and of molecules – the universe follows the laws of classical physics. However, in the quantum world – the impossibly small realm of subatomic particles such as

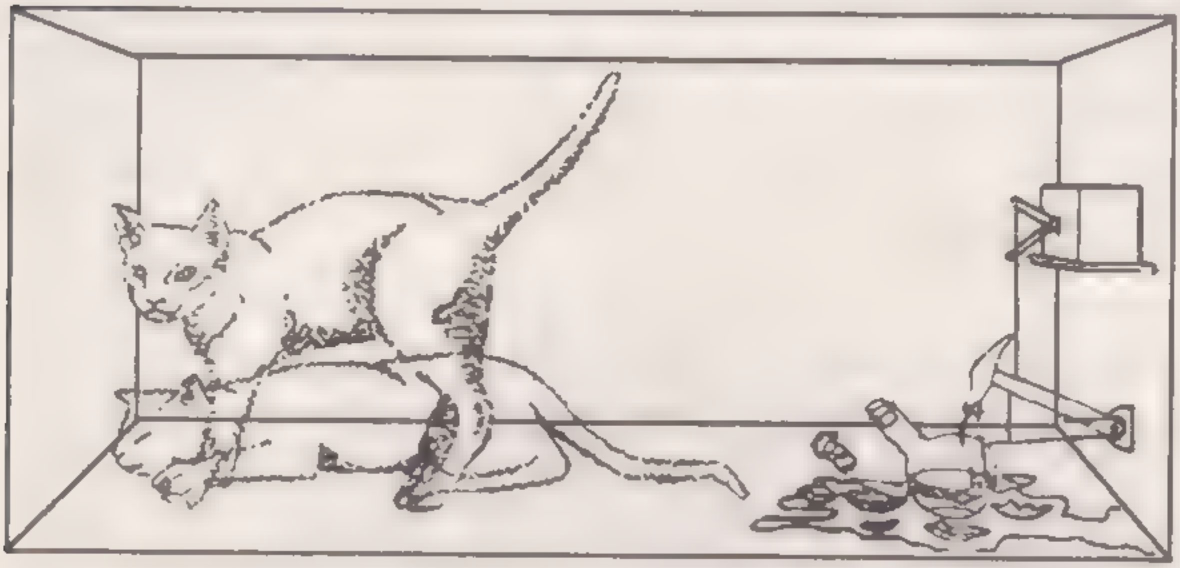
quarks, photons, electrons, etc., a different set of rules apply leading to astounding paradoxes. Without this theory, there would no such thing as nuclear reactors nor laser readers. There would be no way to explain the brilliance of the sun or the functioning of DNA.

Niels Bohr (above left) with Max Planck, two fathers of quantum physics, in a photograph taken in 1930

The cat that was neither dead nor alive

In a quantum physics seminar held in 1958, Bohr gave his opinion on the proposition of one of the speakers as follows: "We all agree that his theory is crazy. The question that divides us is whether it is crazy enough that it could have a chance of being correct." How crazy is quantum mechanics, really? By way of example, let's take the principle of the superposition of states. A particle presents a superposition of states when it occupies more than one position at the same time, or when it simultaneously possesses different quantities of energy. However, when an observer measures the particle it will always be seen to adopt one position or

to possess a specific quantity of energy. Schrodinger himself devised a thought experiment, "Schrodinger's cat," to illustrate this apparently ridiculous notion. Imagine a cat is placed in a sealed, opaque box. Inside the box there is also a flask containing a noxious gas that is connected by some device to a radioactive particle so that, if the particle decays, the gas escapes from the container, and the cat is poisoned. The particle in question has a 50% probability of decaying during a determined period of time. The whole set up, depending as it does on the behaviour of a single particle, is subject to the laws of quantum physics.



Schrodinger's cat is a thought experiment that illustrates the quantum theory concept of the superposition of states.

Let us suppose that the determined period of time has passed. The question is, Is the cat alive or dead? Or, in the jargon of quantum mechanics, what is the state of the box-cat-system? The answer to the question is that, until the observer opens the box and "measures" the state of the system, the particle may or may not have disintegrated and, therefore, there is a system of superposed states: the cat is, strictly speaking, neither alive nor dead, but both simultaneously.

For all those who consider the superposition of states to be a far-fetched hypothesis, it is important to note that alternate interpretations have been proposed by respected physicists. For example, the theory known as *interpretation of possible worlds* maintains that the notion of the superposition of states is an unsustainable

thesis and that what occurs in reality is that, for each of the possible states a particle may find itself in – position, quantity of energy, etc – there exists an alternative universe where the particle adopts one specific state. In other words, in one universe the cat in the box is alive, and in another, dead. When the observer opens the box and verifies that our feline friend is in fact alive, he does so as an integral part of only one of the possible universes. In another parallel universe – complete with its own stars, planets, train stations and ants – this same observer looks inside the box and verifies, undoubtedly with some sadness, that the cat has succumbed to the deadly poison. The supporters of the interpretation of possible worlds still haven't clarified how these universes interact with each other. Even so, what the theory shows is that it is the interpretation of why quantum reality behaves in this way that is in question, not the behaviour itself, which has been confirmed in numerous conclusive experiments.

From bit to qubit

What, however, is the relation between the superposition of states of particles and computation – let alone cryptography? Until 1984 nobody would have even thought to propose a relationship between the two fields. Around that time, the British physicist, David Deutsch, began to throw around a revolutionary idea: what would computers be like if, instead of submitting to the laws of classical physics, they obeyed instead those of quantum mechanics? How could computing benefit from the superposition of states of particles?

Let us recall that conventional computers handle minimal units of information, called bits, capable of assuming opposing values 0 and 1. A quantum computer, on the other hand, could take as its smallest unit of information a particle that presents two possible states. For example, the spin of an electron can only be in one of two directions, up or down. This particle would have the fantastic property of representing the value 0 (spin down), or the value 1 (spin up). Through the superposition of spin states, it could represent the two values simultaneously. This new unit of information has been called a qubit, a contraction of quantum bit, and its manipulation [illegible] opens the doors to a world of super-powerful computers.

A conventional computer performs its calculations sequentially. Let us take as an example the numeric information contained in 32 bits. With this number of bits we can encrypt numbers from 0 to 4,292,967,295. If a conventional computer had to find a specific number within that group, it would have to do so bit by bit

However, a quantum computer could perform the task much faster. To illustrate how, imagine we can put 32 electrons in a special container and make them enter a superposition of states. Then, by applying electromagnetic pulses strong enough to change the spin of an electron from up to down, these 32 electrons—the qubits of our quantum computer—would represent all the possible combinations of spin up (1) and spin down (0) simultaneously. As a result, the search for the desired number would be performed on each and every one of the possible options at once. If we increase the quantity of *qubits* to, say, 250, the number of simultaneous operations that could be performed would be about 10[illegible], a little more than the number of atoms that our universe is thought to contain.

The work of Deutsch proved that quantum computers were a theoretical possibility. That they become a practical reality one day is the objective of dozens of institutions and research groups throughout the world. So far, however, they have not been capable of overcoming the technical difficulties of building a viable quantum computer. Some experts believe it will take another 15 or 25 years to achieve it; others doubt that it is even possible.

A BIG BROTHER FOR THE 21st CENTURY

[illegible] computer waiting to be placed in full operation to change our lives forever?

GOODBYE, DES, GOODBYE

Two years after Shor demonstrated that a quantum computer could conquer the RSA cipher, another American, Lov Grover, did the same with another mainstay of modern cryptography, the DES algorithm. Grover designed a program that allowed a quantum computer to find the correct cipher value from a list of possible values in a time that is the square root of the time it would have taken a conventional computer. Another commonly used algorithm that would be affected by this innovation is the RC5, the standard used by Microsoft web browsers.

The end of cryptography?

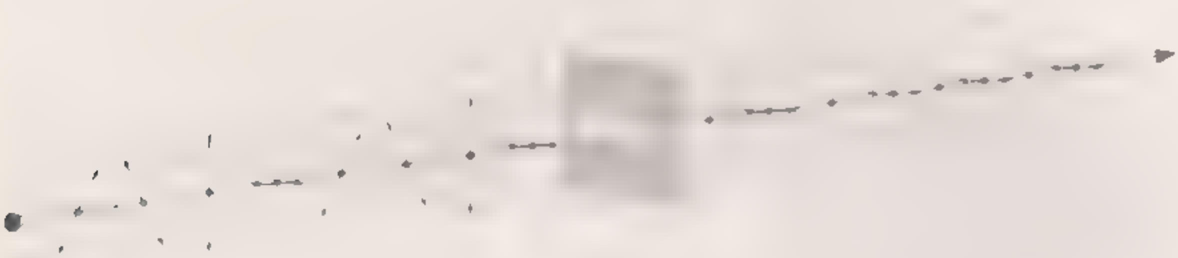
Quantum computing would lead to the death of cryptography as we know it. Let's take as an example the star of modern encryption algorithms, RSA. As we recall, whoever tries to crack an RSA code by brute force will have to successfully factorise the product of two very large prime numbers. This operation is extraordinarily laborious and so far no mathematical shortcut has been found to solving it. Could a quantum computer take on the challenge of factoring a prime number of the size handled by RSA codes? Peter Shor, the American computer scientist, answered affirmatively in 1994. Shor designed an algorithm executable by a quantum computer, and capable of breaking down enormous numbers in infinitely less time than a more powerful conventional computer.

If this astounding device were ever to be constructed, Shor's algorithm would demolish, piece by piece, the powerful cryptographic infrastructure built around RSA and, overnight, the most secret information on the planet would be exposed to the light of day. All contemporary encryption systems would follow the same path. Paraphrasing Mark Twain, we could say that the reports of the death of cryptanalysis have been "greatly exaggerated."

What quantum mechanics takes away, quantum mechanics gives back

One of the foundations of quantum mechanics is called the Uncertainty Principle, formulated by Werner Heisenberg in 1927. Although its exact formulation is very technical, its own creator dared to summarise it as follows: "In principle

we cannot know the present in all its detail." More concretely, it is impossible to determine with any degree of accuracy certain complementary properties of a particle at any given moment. Let us take, for example, the case of light particles (photons). One of their fundamental characteristics is their polarisation, a technical term that refers to the oscillation or vibration of an electromagnetic wave. [Although photons vibrate in all directions, for the purpose of this brief exposition we will assume that they vibrate in four: vertical (↕), horizontal (↔), diagonal to the left (⤡) diagonal to the right (⤢).] Well, then, Heisenberg's principle states that the only way to verify the polarisation of a particular photon is by passing it through a filter or "slit" that in turn can be either horizontal, vertical or diagonal to the left or right. The photons polarised horizontally will pass the horizontal filter unchanged, while those that are polarised vertically will be blocked. As for the photons that are diagonally polarised, half of them will pass through the filter with their polarisation changed to horizontal, and the other half will rebound, completely at random. Furthermore, once a photon is emitted from the filter, it is not possible to know with certainty what its original polarisation was.



If we pass a series of photons of different polarisations through a horizontal filter, [illegible] of those oriented diagonally pass through the filter with their polarisation [illegible]

What is the relationship between the polarisation of photons and cryptography? Very substantial, as we shall see below. To begin with, we will assume the role of a researcher who wants to know the polarisation of a series of photons. To do this he has no other option but to select a filter with a fixed orientation, such as horizontal. Let's suppose that a photon passes through the filter. What information does the researcher get from this? Of course, he can assume that the original polarisation of the photon was not vertical. Can he make any other assumption? No. At first one could think that there are more probabilities that the original photon was oriented

horizontally than diagonally, because half of the diagonals would not pass through the filter. However, the number of diagonally oriented photons is also double the number that are horizontal. It is important to emphasise that the difficulty in detecting the polarisation of a photon is not the result of some technological or theoretical shortcoming capable of being rectified in the future; it is a consequence of the nature of subatomic reality itself. If appropriately exploited, this property can be used to build a completely unbreakable code, the Holy Grail of cryptography.

The indecipherable cipher

In 1984, the American Charles Bennett and the Canadian Gilles Brassard conceived the idea of an encryption system based on the transmission of polarised photons. The first step consists of the sender and the receiver agreeing on a method to assign a 0 or a 1 to one polarisation or another. In the example here, the assignment of 0 and 1 will be a function of two diagrams or bases of polarisation: the first base, called rectilinear and represented by the symbol +, maps the 1 to the polarisation ↕, and the 0 to the polarisation ↔; the second base, called diagonal and represented by the symbol X, assigns a 1 to the polarisation ⤢ and the 0 to ⤡. For example, the message 0100101011 could be transmitted as follows:

Message	0	1	0	0	1	0	1	0	1	1
Base	X	+	+	X	+	X	X	+	X	+
Transmission	⤡	↕	↔	⤡	↕	⤡	⤢	↔	⤢	↕

If a spy intercepts the transmission, he would need to use a filter with a fixed X orientation:

Original message	⤡	↕	↔	⤡	↕	⤡	⤢	↔	⤢	↕
[illegible]	X									
[illegible]	⤡	⤡ or ⤢	⤡ or ⤢	⤡	⤡ or ⤢	⤡	⤢	⤡ or ⤢	⤢	⤡ or ⤢
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	⤡ or ⤢ ↕ ↕

As we can see, not knowing the original base, the spy is unable to get any relevant information whatsoever from the polarisation detected. Even knowing the scheme of assigning 0 and 1 used by the sender and the receiver, if the former alternates the bases in a random fashion, the spy will be mistaken approximately one-third of the time (the table shows a breakdown of all the sending and receiving combinations possible under the described conditions). However, there is a glaringly obvious problem: the receiver is in no better a position than the spy.

Having reached this point, the sender and the recipient could get round the problem by sending each other the sequence of bases used through some secure medium, such as ciphering with RSA. But then the security of the cipher would be at risk from those hypothetical quantum computers.

To overcome this last obstacle, Brassard and Bennett had to add another subtlety to their method. If the reader recalls, the Achilles' heel of the polyalphabetic ciphers of the family of DeVigenere's square was that the use of short, repeated keys generated a regularity in the cipher that created a small but significant opportunity for the cryptanalyst. What would happen, however, if the key used were a random string of characters longer than the message? And what if, for greater security, every message, however insignificant, were ciphered with a different key? The answer is that we would have an unbreakable cipher.

The first person to suggest the use of the polyalphabetic cipher with a unique key was Joseph Mauborgne. Shortly after World War I, when he was the chief signals officer for the American cryptographic service, Mauborgne imagined a notepad of keys composed of random series of more than 100 characters each, that would be given to the sender and the receiver with instruction to destroy the key used on each occasion and to move on to the following one. This system, known as the one-time pad, is, as we said, unbreakable, and can be demonstrated as such mathematically. In fact, top secret communications between some heads of state are carried out with this method.

If the cipher of the one-time pad is so secure, why hasn't its use spread? Why are we worrying about the power of quantum computers and even mentioning the manipulation of photons?

Leaving aside the logistical difficulties of generating thousands of random single-use keys to cipher the same number of messages, the cipher of the one-time pad presents the same weakness as the other classical encryption algorithms: key distribution, just the thing that modern cryptography has been so eager to resolve.

Base of the sender	Bit of the sender	The sender transmits	Detector of the receptor	Is the detector correct?	The receptor detects	Bit of the receptor	Is the bit of the receptor correct?
DIAGONAL	1	↗		No	↕	1	Yes
					↔	0	No
				Yes	↗	1	Yes
	0	↘		No	↕	1	No
					↔	0	Yes
				Yes	↘	0	Yes
RECTOLINEAR	1	↕		Yes	↕	1	Yes
				No	↘	0	No
					↗	1	Yes
	0	↔		Yes	↔	0	Yes
				No	↗	1	No
					↘	0	Yes

However, the transmission of information by polarised photons is the perfect channel for submitting a unique key without danger. For this to occur, three steps prior to transmitting the message are necessary:

1. First, the despatcher sends the receiver a random sequence of 1 and 0 by means of different, equally random, filters of vertical (↕), horizontal (↔) and diagonal (↘, ↗) alignment.

2. The receiver goes on to measure the polarisation of the received photons by the random alternation of rectilinear bases (+) and diagonal bases (X). Since he does not know the sequence of filters used by the sender, a large part of the sequence of 0 and 1 will also be wrong.

3. Finally, the sender and the receiver make contact in whatever manner they prefer, without needing to worry that it is an insecure channel, and they exchange the following information: first, the sender explains what base, rectilinear or diagonal, must be employed to read each photon correctly, but without revealing its polarisation (that is, the filter used). For his part, the receiver tells

BABEL'S MESSAGE

The Argentinian writer [illegible] library so [illegible] the refutations [illegible] A cryptanalyst [illegible] word meant [illegible] others of the same length, and so on to infinity

him in what cases he got the base selection right. As we can see in the preceding table, if a sender and a receiver get their respective bases right, we can be certain that the transmission of 0 and 1 has been completed correctly. Lastly, and in private, they each throw away the bits that correspond to the photons that the receiver identified with the mistaken base.

The result of this process is that the sender and the recipient now share a sequence of 1 and 0 generated completely randomly: the selection of the polarisation filters used by the sender is random, as is the selection of bases used by the receiver. A modest twelve-bit example of the process described above appears in the following drawing:

Bits of the sender	0	[illegible]	1	[illegible]	1	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Detector of the receiver	+	×	×	+	×	+	[illegible]	[illegible]	+	[illegible]	[illegible]	[illegible]
The receiver detects	1	1	1	0	1	0	1	[illegible]	[illegible]	[illegible]	[illegible]	1
Bits kept		1		0	1	[illegible]		[illegible]	0	0	0	1

Take note of the fact that of the bits finally retained, some are discarded even though they were correctly interpreted. This is done because the recipient cannot be certain of having detected them correctly, having used the wrong bases. If the initial transmission consists of a sufficient number of photons, the sequence of 1 and 0 will be long enough to constitute a one-time pad cipher capable of ciphering messages of a normal length.

Let us now put ourselves in the place of a spy who has intercepted both the sent photons and the public conversations of the sender and the receiver. We have already seen that, without knowing exactly what polarisation filter was used by the sender of the message, it is impossible to determine when we have detected the correct polarisation. Nor is the information exchanged by the sender and the receiver of any help, because they never transmit information on the specific polarisations.

What is even more frustrating to the spy, when not having hit upon the correct base and therefore having altered the polarisation of the photon, his intrusion will be revealed – and there is nothing he can do to stay undetected. It is enough for a sender and a receiver to verify a sufficiently long part of the key to detect any manipulation of the polarisation of the photons by an eavesdropper.

To this end, the sender and the recipient agree on a very simple verification protocol. Having completed the three preliminary steps specified above, and with enough retained bits, the sender makes contact with the receiver, again by some conventional medium, and together they check a group (let's say 100) of bits chosen at random from the total. If the 100 match, both the sender and the receiver can be completely certain that no spy has snooped on the transmission, and they can consider the sequence to be a good one-time cipher. Otherwise, the sender and the receiver have to start the process all over again.

32cm of absolute secrecy

Brassard's and Bennett's method is impeccable from a theoretical point of view but when the theory was eventually put into practice, it was met with a great deal of scepticism. In 1989, following more than a year of hard work, Bennett fine-tuned a system consisting of two computers separated by a distance of 32 centimetres, one of which would act as the sender, and the other as the receiver. After several hours of trials and adjustments, the experiment was declared a success. The sender and the receiver completed all the stages of the process, and were even able to verify their respective ciphers. Quantum cryptography was possible.

Bennett's historic experiment had the obvious flaw of only sending secrets less than the length of a pace – a whisper would have probably been just as effective. However, in following years, other research teams increased the reach of the transmission. In 1995, researchers at the University of Geneva used an optical fibre to send messages 23 km. In 2006, a team from the Los Alamos National Lab in the United States, reached 107 km with the same procedure. Although they are not yet of a sufficient distance to be useful in conventional communications, they can be employed on small scales in places where the utmost secrecy is paramount, such as government buildings and company headquarters.

Leaving aside considerations relating to the physical restriction of sending messages, there is no possibility that the transmission be sabotaged, even at the quantum level. This quantum code represents the final triumph of secrecy over indiscretion, of cryptography over cryptanalysis. All we have to concern ourselves with now – not a minor issue by any means – is how to apply this powerful tool and who will get the benefit.

Appendix

Various classic ciphers – and a hidden treasure

Below, we will set out various classic cryptographic ciphers mentioned in the main chapters, but not developed in depth there. All of them are representative of a variety of cryptographic techniques, or are interesting simply as diversions. We end the selection of classical ciphers with a fictional decryption by the American writer Edgar Allen Poe, which illustrates frequency analysis perfectly.

Polybius's cipher

This cipher, one of the oldest for which we have detailed information, is based on selecting five letters of the alphabet to act as the row and column headings outside of a five-by-five grid, and then filling the grid with the letters of the alphabet. The cipher consists of having each letter correspond to the pair of letters indicated by the rows and the columns of the table. Originally the Greek alphabet of 24 letters was used, so I and J of the English 26-letter alphabet are usually combined (see grid below, which, for simplicity, uses A–E as the headings). The grid is filled in an order agreed upon by the sender and the receiver. Now let's examine the following table:

	A	B	C	D	E
A	A	B	C	D	E
B	F	G	H	I	K
C	L	M	N	O	P
D	Q	R	S	T	U
E	V	W	X	Y	Z

Note that the ciphered alphabet has to be 25 letters (5 × 5). The ciphered alphabet can also be organised according to numeric values (for example, the numbers 1, 2, 3, 4 and 5). In that case the table could be:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Let's look at an example of Polybius's cipher using the two versions. The plaintext message is "BLANKS." From the first table we get:

B will be substituted by the pair AB.
L will be substituted by the pair CA.
A will be substituted by the pair AA.
N will be substituted by the pair CC.
K will be substituted by the pair BE.
S will be substituted by the pair DC.

The ciphered message is "ABCAAACCBEDC." If we use the numeric version, from an analogous process, we get: 123111332543.

Gronsfeld's cipher

This cipher, invented by the Dutchman Jost Maximilian Bronckhorst, the Count of Gronsfeld, was used in Europe in the 17th century. It is a polyalphabetic cipher, analogous to De Vigenere's square, but less difficult (and secure). To encrypt a message we start with the following table:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0:	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
1:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
2:	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
3:	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
4:	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
5:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
6:	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
7:	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
8:	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
9:	[illegible]	D	E	F	G	H	[illegible]	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Next, we select a number randomly from [illegible] in the message we wish to cipher. If the plaintext is [illegible] MATH[illegible] [illegible] pick 12 numbers randomly, for example 1, 2, 3, 4, 5, [illegible] 2. [illegible] numbers will be the key of the cipher. Next [illegible] letter corresponding to the row [illegible] page.

Message	M	A	T	H	E	[illegible]	A	T	[illegible]	[illegible]	[illegible]	[illegible]
Key	1	2	3	4	5	[illegible]	7	8	[illegible]	[illegible]	[illegible]	[illegible]
Ciphered message	P	F	A	S	R	D	T	Q	[illegible]	[illegible]	[illegible]	[illegible]

M is ciphered as P, taken from the letter on row 1 of the M column, and so on. The whole message is ciphered as PFASRDTQKEDQ. The letter A of the message is ciphered as F, T, and D. As is the general case of polyalphabetic ciphers, this encryption system is resistant to brute force and frequency analysis. The number of keys in a Gronsfeld's Cipher for an alphabet of 26 letters is $26! \times 10$ [illegible] $4.03 \times 10^{[illegible]}$ keys.

The Playfair cipher

The creators of this cipher, Baron Lyon Playfair and Sir Charles Wheatstone (also the pioneer of the electric telegraph) were [illegible], and shared a love of cryptography. The method [illegible] antecedent, Polybius's cipher, and also employs a table of [illegible] rows and [illegible] columns. In a first step, each character of the plaintext is substituted [illegible] according to a cipher of 5 different letters. In our example, the [illegible] JAMES. In the case of a 26 character alphabet, we generate [illegible].

J	A	M	E	S
B	C	D	F	[illegible]
H	I-K	L	[illegible]	[illegible]
P	Q	R	[illegible]	[illegible]
V	W	X	[illegible]	Z

Next, the plaintext message is divided into pairs of letters or *digraphs*. The two letters of all the digraphs have to be different, and to avoid potential coincidences we use the letter X. We also use this letter to complete a digraph in case the last letter is alone.

For example, for the plaintext message "TRILL", the digraph division is:

TR IL Lx.

The word "TOY" is broken down as:

TO Yx.

Once we have the plaintext message in digraph form, we can begin to cipher it, taking into account three prerequisites:

a) That the two letters of the digraph are in the same row
b) That the two letters of the digraph are in the same column
c) None of the above.

In the case of (a), the characters of the digraph are replaced by the letter located to the right of each one (the "next one" in the natural order of the table). In this way, the pair JE is ciphered as AS:

J	A	M	E	S

In the case of (b), the characters of the digraph are replaced by the letter that is located immediately below in the table. For example, the digraph EF is ciphered as FY, and TY as YE:

E
F
N
T
Y

In the case of (c), to cipher the first letter of the digraph we look at its row until we reach the column that contains the second letter; the cipher of the plaintext is that found at the intersection of the two. To cipher the second letter, we look at its row until we reach the column that contains the first letter; the cipher of the plaintext is, again, that found at the intersection.

For example, in the digraph CO, the C is ciphered as G and the O as an I or a K.

J	A	M	E	
B	C	D	F	
H	K		N	O
P	Q	R	T	
V	W	X	Y	

To cipher the message "TEA" with the keyword JAMES we [illegible] as follows:

- We express it in digraph: TE Ax.
- The T is ciphered with a Y.
- The E as an F.
- The A as an M.
- The X as a W.

The ciphered message is "YFMW".

The cryptogram of *The Gold-Bug*

William Legrand, the protagonist of *The Gold-Bug* (1843) by Edgar Allan Poe, discovers where a fabulous treasure [illegible] ten on a piece of parchment. The [illegible] method based on the frequency of [illegible] English text. The ciphered message is as follows:

```
53‡‡†305))6*.4826)4‡ [illegible]
.1‡(.:‡*8†83(88)5*†.46[illegible]
5*†2:*‡(;4956*2(5*4)8¶8*.4069285[illegible]
‡.1(‡9:48081;8:8‡1;48†85.4)485[illegible]
‡9;48;(88;4(‡?34;48)4‡.161.[illegible]
```

Legrand starts with the assumption that the original text was written in English. The letter that occurs most frequently in English is e. Next, and in order of appear

ance from most to least frequent, we have the letters *a, o, i, d, h, n, r, s, t, u, y, c, f, g, l, m, w, b, k, p, q, x, z*.

Our hero creates a table from the cryptogram. In the first row, the characters of the ciphered message, and in the second, the frequency of their appearances.

8	;	4	‡	)	*	5	6	(	†	1	0	9	2	:	3	?	¶	[illegible]
33	26	19	16	16	13	12	11	10	8	8	6	5	5	4	4	3	2	1

Therefore, 8 is very probably the letter *e*. Next he looks for appearances of the trio of characters *the*, also very common, which allows him to translate the characters ;, 4, and 8.

The appearance of the term ";(88", now that he knows that it represents "t(ee" lets him deduce that the missing term (can only be an *r* given that *tree* is the best possibility in the dictionary. Finally, thanks to similar ingenious cryptanalytic techniques and with a great deal of patience, he arrives at the following ciphered partial alphabet:

5	†	8	3	4	6	*	‡	(	;	?
A	d	e	g	h	i	n	o	r	t	u

That is enough to decipher the message:

"A good glass in the bishop's hostel in the devil's seat
forty-one degrees and thirteen minutes northeast and by north
main branch seventh limb east side shoot from the left eye of the death's-head
a bee line from the tree through the shot fifty feet out."

Prime numbers and their value in cryptography

Real mathematics has no effects on war. No one has yet discovered any warlike purpose to be served by the theory of numbers.
Godfrey H. Hardy, *A Mathematician's Apology* (1940)

In order to decrypt a message it is essential that the cipher have an inverse. As we have already observed in the study of affine codes, a way to guarantee this property is to work with a prime number modulus. Moreover, the product of prime numbers constitutes an irreversible function, that is to say, once the multiplication has been performed, it is a very laborious task to ascertain the value of the original factors.

This property makes this operation a very useful tool for systems based on asymmetrical keys, like the RSA algorithm, that in turn constitute the basis for public key cryptography. Here is a more detailed look at the overlap between prime numbers and cryptography, and we will demonstrate what we learn through the formal mathematical operation of RSA.

Prime numbers and the "other" Fermat theorem

Prime numbers as a group are a subset of the natural numbers that includes all the elements of the bigger set that are larger than 1 and only divisible by themselves and by one. A fundamental theorem of arithmetic establishes that any natural number larger than one can always be represented as the product of the powers of prime numbers, and this representation is unique. For example:

$$20 = 2^2 \cdot 5$$
$$63 = 3^2 \cdot 7$$
$$1{,}050 = 2 \cdot 3 \cdot 5^2 \cdot 7.$$

All prime numbers except for 2 are odd. The only two consecutive prime numbers are 2 and 3. Odd consecutive prime numbers, that is, those that are just 2 apart (for example, 17 and 19), are called *twin prime numbers*. Mersenne and Fermat primes are also of particular interest.

A prime number is Mersenne prime if, when added to 1, the result is a power of 2. For example, 7 is a Mersenne prime number since $7 + 1 = 8 = 2^3$,

The first eight Mersenne prime numbers are therefore:

3; 7; 31; 127; 8,91; 131,071; 524,287; 2,147,483,647

Today we know of only 40 or so Mersenne prime numbers. The largest is a gigantic number $2^{[illegible]} - 1$, discovered in 2008. By way of comparison, the estimated number of elemental particles in the entire universe is less than $2^{[illegible]}$.

For his part, Fermat's prime number is a prime number in the form of

$F_n = 2^{2^n} + 1$, with n being a natural number.

We only know five Fermat primes: 3 ($n=0$), 5 ($n=1$), 17 ($n=2$), 257 ($n=3$) and 65,537 ($n=4$).

Fermat's primes carry the name of the illustrious French jurist and mathematician who discovered them: Pierre de Fermat (1601–1665). The Frenchman made numerous and important additional discoveries relative to prime numbers. One that stands out is Fermat's little theorem, which establishes that:

If p is a prime number, then for any integer $a^p \equiv a$ in mod p.

That result is of great importance in modern cryptography, as we shall now see.

From Euler to RSA

Another result of great interest in modular arithmetic is that known as Bézout's identity. The identity establishes that if a and b are positive integers, the equation $\gcd(a, b) = k$ is equivalent to there being two whole numbers, p, q that satisfies:

$$pa + qb = k.$$

In the particular case, that $\gcd(a, b) = 1$, we can claim that there are whole numbers p and q such that

$$pa + qb = 1.$$

If we work in modulus n, we can establish that if [illegible] are, necessarily, whole numbers p and q such that $pa + qb = 1$. [illegible] of modulus n we hold that $qn = 0$ from which we conclude that [illegible] $pa = 1$, that is, the inverse of a in modulus n exists and is p.

The number of elements with an inverse in modulus n will be [illegible] of natural numbers a smaller than n that fulfil $\gcd(a,n) = 1$. This [illegible] is known as Euler's Formula and is denoted as $\varphi(n)$.

If the factorisation of n in prime numbers is $n = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$, then

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right) \cdots \left(1 - \frac{1}{p_k}\right)$$

If, for example, $n = 1{,}600 = 2^6 5^2$ we will have:

$$\varphi(1{,}600) = 1{,}600\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right) = 640.$$

Fine tuning even more, if the situation is that n is prime, we get that for any value of a the $\gcd(a,n) = 1$ and, consequently, any value of a will have an inverse modulus n, and, therefore, $\varphi(n) = n - 1$.

Let us take a moment and [illegible] we have reached so far:

1) $\varphi(n)$ is called Euler's Formula [illegible] than n that are prime with n.

2) If $n = pq$ with p and q being two prime numbers, then

$$\varphi [illegible]$$

3) From Fermat's little theorem, we know [illegible] greater than 0 and p is a prime number we will [illegible] $\equiv a$ (mod. p) that is the same as affirming that $a^{p-1} \equiv 1$ (mod. p).

All that's left is to add the final piece, which is provided by Euler's Formula. Euler affirms that:

4) If $\gcd(a,n) = 1$, then we verify the equation [illegible] $\equiv$ [illegible] mod n

Why does the RSA algorithm work?

Armed with the knowledge expressed above, we are ready to show the mathematical arguments that underlie the ciphering process of the RSA algorithm.

The algorithm in question encrypts a numerical representation m of any message whatsoever, with p and q, two prime numbers, and $n = p \cdot q$. We call e any value that verifies that $\gcd(e, \varphi(n)) = 1$ and we call d the inverse of n in modulus $\varphi(n)$ [that we know exists since $\gcd(e, \varphi(n)) = 1$]. So:

$$d \cdot e = 1 \text{ mod. } (n).$$

The ciphered message, M, is ciphered according to $M = m^e \text{ mod } n$. The algorithm presupposes that the original message m is obtained by $m = M^d = (m^e)^d$ (mod n). Verifying this equation is equivalent to demonstrating the validity of RSA. To do this, we combine Fermat's theorem with Euler's formula.

Let's consider two cases:

1) If $(m,n) = 1$ according to Euler's formula $m^{\varphi(n)} \equiv 1$ (mod. n).

We start from the relation that is equivalent to $e \cdot d - 1 \equiv 0$ (mod. $\varphi(n)$), that is, there is a value k, whole, such that $e \cdot d - 1 = k \cdot \varphi(n)$, that is, $e \cdot d = k \cdot \varphi(n) + 1$. With this, applying Euler's formula, we have the equation:

$$(m^e)^d = m^{ed} = m^{k\varphi(n)+1} = m^{k\varphi(n)} \cdot m = (m^{\varphi(n)})^k \cdot m \equiv 1^k \cdot m \text{ (mod. } n) =$$
$$= m \text{ (mod. } n).$$

This is the result we were seeking.

2) If $\gcd(m,n) \neq 1$, and $n = p \cdot q$, m will contain as factor only p, only q, or both simultaneously.

In the first case:

a) m will be a multiple of p, that is, there is a whole number r such that $m = r \cdot p$. Therefore $m^{de} \equiv 0$ (mod. p), and finally $m^{de} \equiv m$ (mod. p), in other words, there is a value of A so that:

$$m^{de} - m = \mathrm{A}p. \qquad (1)$$

In the second case,

b) we have that

$$(m^e)^d = m^{ed} = m^{k\ \ (n)+1} = m^{k\ \ (n)} \cdot m = (m^{\ \ (n)})^k \cdot m \equiv$$
$$\equiv (m^{(q-1)})^{k(p-1)} \cdot m \ (\mathrm{mod.}\ n) = m.$$

Since $\gcd(m,n) = p$ the $(m,q) = 1$ and by Fermat's theorem $m^{(q-1)} \equiv 1$ (mod. q).

Applied to the initial equation:

$$(m^e)^d = m^{ed} = m^{k\ \ (n)+1} = m^{k\ \ (n)} \cdot m = (m^{\ \ (n)})^k \cdot m \equiv$$
$$\equiv (m^{(q-1)})^{k(p-1)} \cdot m \ (\mathrm{mod.}\ n) = 1^{k(p-1)} m = m \ (\mathrm{mod.}\ q).$$

from which we conclude that there is a value of B such that

$$m^{de} - m = \mathrm{B}q. \qquad (2)$$

From expressions (1) and (2) we [illegible], therefore $m^{de} - m \equiv 0$ (mod. n).

The process is analogous if we consider q. In the case where [illegible] is the multiple of both p and q simultaneously, the result is trivial. Consequently,

$$(m^e)^d \equiv m \ (\mathrm{mod.}\ n).$$

The cipher of the RSA algorithm is thus demonstrated

Bibliography

FERNANDEZ, S., *Classical Cryptography*, [illegible]

GARFUNKEL, S., *Mathematics [illegible]*, Madrid, [illegible] UAM, 1998.

GOMEZ, J., *From the [illegible]* [illegible]

KAHN, D., *The Codebreakers: The Story of Secret Writing*, New York, [illegible]

SINGH, S., *The Secret Codes*, Madrid, Editorial Debate, 2000.

TOCCI, R., *Digital Systems: Principles and Applications*, Prentice Hall, 2[illegible]

Index

Mathematicians, Spies and Hackers

Coding and encryption

The safety and confidentiality of communication in the digital world depends on a complex code designed by mathematics. This book offers a stimulating journey through the arithmetic of security and secrecy, introduces you to the encryptors and decryptors who determined the destiny of nations and uncovers the language through which computers communicate.